Alexander Denisjuk

Matematyka Dyskretna

Skrypt przeznaczony dla studentów
kierunku Informatyka Stosowana

Prywatna Wyższa Szkoła Zawodowa w Giżycku

Giżycko 2004

I. Kombinatoryka

Oznaczenie. Niech X będzie skończonym zbiorem. Przez $|X|$ oznaczamy *ilość elementów* X

Twierdzenie 1.1. *Niech X i Y — dwa skończone zbiory. Wówczas $|X|=|Y|$ wtedy i tylko wtedy, gdy istnieje wzajemnie jednoznaczne odwzorowanie $f\colon X \to Y$.*

Wszystkie zbiory w tym rozdziale będą *skończone.*

1. Zasady sumy i iloczynu

Twierdzenie 1.2. *Niech X_1 i X_2 będą dwa zbiory nieprzecinające się, $X_1 \cap X_2 = \emptyset$. Wtedy $|X_1 \cup X_2| = |X_1| + |X_2|$.*

Twierdzenie 1.3. *Niech X_1 i X_2 będą dwa zbiory. Wtedy*

$$|X_1 \times X_2| = |X_1| \cdot |X_2|.$$

Twierdzenie 1.2′. *Niech $X_1, X_2, \dots, X_k$ będą zbiory nieprzecinające się parami, $X_i \cap X_j = \emptyset$ dla $i \neq j$. Wtedy*

$$\Bigl|\bigcup_{i=1}^{k} X_i\Bigr| = \sum_{i=1}^{k} |X_k|.$$

Twierdzenie 1.3′. *Niech $X_1, X_2, \dots, X_k$ będą zbiory. Wtedy*

$$\Bigl|\prod_{i=1}^{k} X_i\Bigr| = \prod_{i=1}^{k} |X_k|.$$

2. Wariacje i kombinacje

Definicja 1.4. Zestaw elementów $x_{i_1}, \dots, x_{i_1}$ zbioru $X = \{x_1, \dots, x_n\}$ nazywa się (n,r)-*próbką.* Jeśli kolejność elementów w próbce jest istotną, to próbka jest *uporządkowana,* inaczej — *nieuporządkowana.* Jeżeli pośród elementów próbki dopuszczalne są jednakowe, to próbka jest *z powtórzeniami,* inaczej — *bez powtórzeń.* Uporządkowana (n,r)-próbka nazywa się (n,r)-*wariacją.* Nieuporządkowana (n,r)-próbka nazywa się (n,r)-*kombinacja.* (n,n)-wariacja bez powtórzeń nazywa się *permutacją* zbioru X.

Ilość (n,r)-wariacji z powtórzeniami oznacza się przez $\bar{A}_n^r$, bez powtórzeń — przez A_n^r, ilość permutacji n-elementowego zbioru — przez P_n. Ilość (n,r)-kombinacji z powtórzeniami oznacza się przez $\bar{C}_n^r$, bez powtórzeń — przez C_n^r (lub $\binom{n}{r}$).

Twierdzenia 1.5.

1. $\bar{A}_n^r = n^r$.
2. $A_n^r = \frac{n!}{(n-r)!}$ *dla* $r \leqslant n$ *i* $A_n^r = 0$ *dla* $r > n$.
3. $P_n = n!$.
4. $C_n^r = \frac{A_n^r}{r!} = \frac{n!}{r!(n-r)!}$ *dla* $r \leqslant n$ *i* $C_n^r = 0$ *dla* $r > n$.
5. $\bar{C}_n^r = C_{n+r-1}^r$.

3. Wariacje i odwzorowania

Definicja 1.6. Przez Y^X oznaczamy *zbiór odwzorowań z* X w Y.

Twierdzenia 1.7. *Niech* $|X| = r$, $|Y| = n$. *Wtedy*

1. $|Y^X| = \bar{A}_n^r = n^r = |Y|^{|X|}$.
2. *Ilość wszystkich różnowartościowych odwzorowań* $f : X \to Y$ *równa jest* A_n^r.
3. *Ilość wzajemnie-jednoznacznych odwzorowań* n*-elementowego zbioru w siebie równa jest* $n!$.

4. Przykłady stosowania wzorów

Przykład 1.8. Na ile sposobów można pomalować kwadrat, podzielony na cztery części pięcioma kolorami a)[1] dopuszczając jednakowe kolory; b)[2] jeśli różne części maluje się na różne kolory?

Przykład 1.9[3]. Ile jest sposobów na wybranie 20 numerów z 80?

Przykład 1.10. W ilu przypadkach w grze w «Multi lotka» (zgadywanie 5 numerów) zostaną prawidłowo wybrane a)[4] dokładnie 3 numery; b)[5] dokładnie 4 numery; c)[6] dokładnie 5 numerów; d)[7] nie mniej niż 3 numery?

Przykład 1.11[8]. Z talii kart, liczącej 52 karty, wybrano 10 kart. W ilu przypadkach spośród wybranych okażą się wszystkie cztery asy?

Przykład 1.12[9]. Zestaw liczy 30 monet wartości 1, 2 oraz 5 złotych. Ile istnieje zestawów?

---

[1] $5^4 = 625$.
[2] $5!/(5-4)! = 120$.
[3] $C_{80}^{20} = 3\,535\,316\,142\,212\,174\,320$.
[4] $C_{20}^3 C_{60}^2 = 2\,017\,800$.
[5] $C_{20}^4 C_{60}^1 = 290\,700$.
[6] $C_{20}^5 = 15\,504$.
[7] $2\,324\,004$.
[8] $C_{48}^6 = 12\,271\,512$.
[9] $\bar{C}_3^{30} = C_{32}^{30} = 496$.

Literatura

1. Rasiowa H.: *Wstep do matematyki wspolczesnej,* PWN, Warszawa, 1975.
2. Lipski W.: *Kombinatoryka dla programistow,* WNT, Warszawa, 1982.
3. Robin J. Wilson: *Wprowadzenie do teorii grafow,* PWN, Warszawa, 1998.
4. Ross K. A., Wright C. R. B.: *Matematyka dyskretna,* PWN, Warszawa 1996
5. Kulikowski J. L.: *Zarys teorii grafow,* PWN, Warszawa 1986.
6. Banachowski L., Diks K., Rytter W.: *Algorytmy i struktury danych,* WNT, Warszawa 1996.
7. *Klin M. C., Poesche R., Rosenbaum K.: Algebra stosowana dla matematykow i informatykow: grupy, grafy, kombinatoryka,* PWN, Warszawa 1996.
8. ШеньА.: *Программирование: теоремы и задачи,* Москва, МЦНМО, `http://www.mccme.ru/ium/ancient/`
9. Ландо С.: *Комбинаторика,* Москва, МЦНМО, 1999, `http://www.mccme.ru/ium/ancient/combs93.htm`

Uznanie

Jestem wdzięczny studentom pierwszego roku (2003-2004), którzy wskazali na liczne błędy wstępnej wersji skryptu oraz znacznie poprawili polszczyznę.

Spis treści

Przykład 1.13[10]**.** Wzór na ilość całkowitych rozwiązań równania

$$\begin{cases} x_1 + \cdots + x_n = r, \\ x_i \geqslant a_i, \quad i = 1, \ldots, n, \end{cases}$$

gdzie $n \geqslant 1$, a_i są liczby całkowite.

Rozwiązanie. $\bar{C}_n^q$ jest liczbą nieujemnych całkowitych rozwiązań równania $y_1 + \cdots + y_n = q$. Każdemu rozwiązaniu podporządkujemy kombinację z powtórzeniami elementów zbioru $\{ b_1, \ldots, b_n \}$, w którą b_1 wchodzi x_1 razy, b_2 wchodzi x_2 razy, … , b_n wchodzi x_n razy.

5. Rozbicia

Definicje 1.14.

1. *Rozbiciem* zbioru X nazywa się zbiór takich podzbiorów $X_1, \ldots, X_k$, $X = \bigcup\limits_{i=1}^{k} X_i, X_i \bigcap X_j = \emptyset$ przy $i \neq j$. (Oznaczenie $X = \coprod\limits_{i=1}^{k} X_i$.)
2. Ilość rozbić zbioru X, $|X| = n$ na k podzbiorów, takich że $|X_i| = n_i$ oznacza się przez $C_n^{n_1,\ldots,n_k}$.
3. Ilość rozbić zbioru X, $|X| = n$ na podzbiory, spośród których dla każdego $i = 1, \ldots n$ jest dokładnie $m_i \geqslant 0$ podzbiorów z i elementami $(\sum\limits_{i=1}^{n} i m_i = n)$ oznacza się przez $N(m_1, \ldots, m_n)$.

Twierdzenia 1.15.

1. $C_n^{n_1,\ldots,n_k} = C_n^{n_1} C_{n-n_1}^{n_2} \ldots C_{n-n_1-\cdots-n_{k-1}}^{n_k} = \frac{n!}{n_1! \ldots n_k!}$.
2. $C_n^{n_1,\ldots,n_k}$ *zgadza się z ilością* (n,k)*-wariacji, wśród elementów których znajduje się* n_1 *elementów* 1*-go typu,* n_2 *elementów* 2*-go typu, … ,* n_k *elementów* k*-go typu.*
3. $N(m_1, \ldots, m_n) = \frac{n!}{m_1! \ldots m_n! (1!)^{m_1} \ldots (n!)^{m_n}}$.

Dowód. (2) Każdej wariacji przyporządkujemy rozbicie zbioru $\{ 1, \ldots, n \}$ na podzbiory $X_1, \ldots, X_k$, gdzie X_i jest zbiorem numerów elementów i-go typu w próbce.

(3) Każde nieuporządkowane rozbicie w $m_1! m_2! \ldots m_n!$ sposobów można sprowadzić do uporządkowanych rozbić postaci

$$X_1, \ldots, X_{m_1}, \;\; X_{m_1+1}, \ldots, X_{m_1+m_2}, \;\; \ldots, \\ X_{m_1+m_2+\cdots+m_{n-1}+1}, \ldots, X_{m_1+m_2+\cdots+m_n},$$

gdzie $|X_1| = \cdots = |X_{m_1}| = 1$, $|X_{m_1+1}| = \cdots = |X_{m_1+m_2}| = 2$, … , $|X_{m_1+m_2+\cdots+m_{n-1}+1}| = \cdots = |X_{m_1+m_2+\cdots+m_n}| = n$.

[10] $$\begin{cases} \bar{C}_n^{r-\sum_{i=1}^n a_i} = C_{n-r-\sum_{i=1}^n a_i-1}^{r-\sum_{i=1}^n a_i}, & \text{jeśli } r \geqslant \sum_{i=1}^n a_i, \\ 0, & \text{jeśli } r < \sum_{i=1}^n a_i. \end{cases}$$

Przykład 1.16[11]**.** Grupa studentów liczy 25 osób. W wyborach samorządowych na P. Nowaka głosowało 12 osób, przeciw niemu — 10 osób, trzy osoby wstrzymały się od głosu. Na ile sposobów mogło odbyć się takie głosowanie?

Przykład 1.17[12]**.** Na ile sposobów można pomalować kwadrat, rozbity na dziewięć części w czterech kolorach tak, żeby w pierwszym kolorze zostały pomalowany 3 części, w drugim — 2, w trzecim — 3, w czwartym — 1?

Przykład 1.18[13]**.** Na ile sposobów można stworzyć 5 koalicji po 5 osób z grupy liczącej 25 osób?

Przykład 1.19[14]**.** Ile jest sposobów określenia na zbiorze $X = \{1, \ldots, 25\}$ relacji równoważności z trzema klasami abstrakcji?

6. Wzór wielomianowy

Twierdzenie 1.20. $(x_1 + \cdots + x_k)^n = \sum\limits_{n_1+\cdots+n_k=n} \frac{n!}{n_1! \ldots n_k!} x_1^{n_1} \ldots x_k^{n_k}$.

Przykład 1.21[15]**.** Ustalić współczynnik przy $x_1^3 x_2^4 x_3^3$ w rozwinięciu $(x_1 + x_2 + x_3)^{10}$.

7. Zasada włączeń i wyłączeń

Twierdzenie 1.22. *Niech X_i będą zbiory, $i = 1, \ldots, n$, $n \geqslant 2$. Wówczas*

$$\begin{aligned}|X_1 \cup \cdots \cup X_n| = \big(|X_1| + \cdots + |X_n|\big) - \\ - \big(|X_1 \cap X_2| + |X_1 \cap X_3| + \cdots + |X_{n-1} \cap X_n|\big) + \\ + \big(|X_1 \cap X_2 \cap X_3| + \cdots + |X_{n-2} \cap X_{n-1} \cap X_n|\big) - \ldots \\ \cdots + (-1)^{n-1} |X_1 \cap \cdots \cap X_n|.\end{aligned}$$

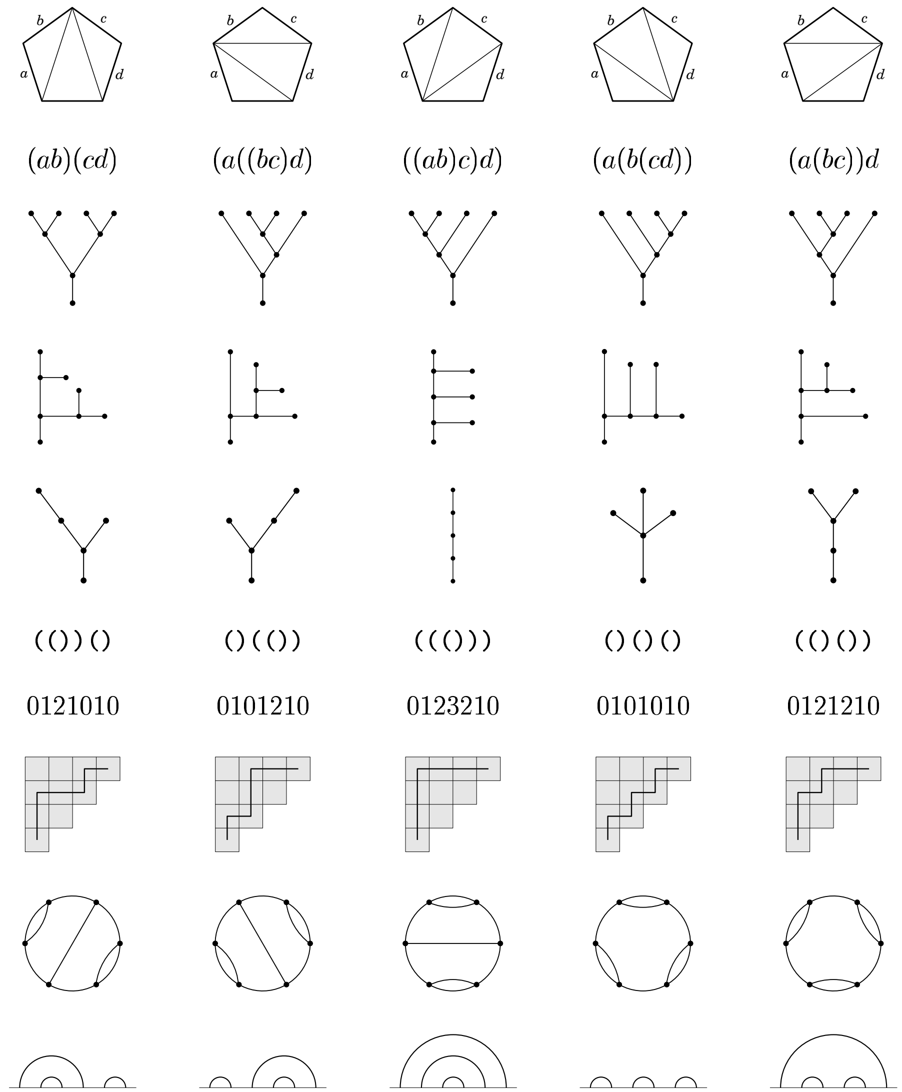
[11] $C_{25}^{12,10,3} = 1\,487\,285\,800$.
[12] $C_9^{3,2,3,1} = 5\,040$.
[13] $N(0,0,0,0,5,0,\ldots,0) = \frac{25!}{5!^6}$.
[14]

$$\sum_{\substack{m1+2m_2+\cdots+25m_{25}=25\\ m1+m_2+\cdots+m_{25}=3}} N(m_1,\ldots,m_{25}) = \\ = \sum_{\substack{m1+2m_2+\cdots+25m_{25}=25\\ m1+m_2+\cdots+m_{25}=3}} \frac{25!}{m_1!\ldots m_{25}!(1!)^{m_1}\ldots(25!)^{m_{25}}}.$$

[15] $C_{10}^{3,4,3} = 4200$.

19. Wypisz pięć pierwszych liczb Catalana.
20. Wypisz funkcję tworzącą dla liczb Catalana.
21. Udowodnij, że dla liczb Catalana $\frac{c_{n+1}}{c_n} = \frac{4n+2}{n+2}$.
22. Podaj przykład zagadnienia z kombinatoryki, w którym występują liczby Catalana.
23. Ile jest sposobów triangulacji wieloboku o n wierzchołkach?
24. Na ile sposobów można wykonać mnożenia w ilozynie $a_1 a_2 \ldots a_n$?
25. Ile istnieje drzew binarnych o n wierzchołkach?
26. Zdefiniuj drzewo binarne.
27. Podaj przykład drzewa binarnego o 10 wierzchołkach.

Skorowidz



Rysunek 6.2. Zagadnienia, związane z liczbami Catalana*

15. Wyprowadzić funkcję tworzącą dla liczb Fibonacciego.
16. Wypisać jawny wzór na k-tą liczbę Fibonacciego.
17. Wyprowadzić jawny wzór na k-tą liczbę Fibonacciego.
18. Zdefiniować liczby Catalana.

* Rysunek jest zapożyczony z materiałów S. Dużyna, Moskiewski Niezależny Uniwersytet, za zgodą autora

Wniosek 1.23. *Nich X będzie zbiorem, X_i, $i = 1, \dots, n$ będą jego podzbiorami. Wówczas*

$$\big|X \setminus (X_1 \cup \dots \cup X_n)\big| = |X| - \big(|X_1| + \dots + |X_n|\big) + \\ + \big(|X_1 \cap X_2| + \dots + |X_{n-1} \cap X_n|\big) - \dots + (-1)^n |X_1 \cap \dots \cap X_n|.$$

Wniosek 1.24. *Niech X będzie N-elementowym zbiorem, $\alpha_1, \dots, \alpha_k$ będą własności (predykaty jednoargumentowe), określone na zbiorze X. Przez*

$$N(\alpha_{i_1}, \dots, \alpha_{i_j})$$

oznaczymy ilość elementów zbioru X, posiadających jednocześnie własności $\alpha_{i_1}, \dots, \alpha_{i_s}$. Przez N_0 oznaczmy ilość elementów X, nie posiadających żadnych z wymienionych własności.

Wtedy

$$N_0 = N - S_1 + S_2 - \dots + (-1)^k S_k,$$

gdzie

$$S_j = \sum_{1 \leqslant i_1 < \dots < i_j \leqslant k} N(\alpha_{i_1}, \dots, \alpha_{i_j}), \quad j = 1, \dots, k.$$

Przykład 1.25[16]**.** Niech $X = \{0, 1, \dots, 10\}$. Obliczyć ilość elementów zbioru X, nie posiadających żadne z wymienonych własności: a) x jest liczbą parzystą, b) $x > 6$, c)$2 < x < 8$.

Przykład 1.26. Wyznaczyć ilość całkowitych rozwiązań układu

$$\begin{cases} x_1 + x_2 + \dots + x_n = r, & \\ a_i \leqslant x_i \leqslant b_i, & i = 1, \dots, n \end{cases}$$

Rozwiązanie. Określimy predykaty $\alpha_i = (x_i \geqslant b_i + 1)$. Zbiór X będzie zbiorem rozwiązań bez górnych ograniczeń.

Przykład 1.27[17]**.** Wyznaczyć ilość trzycyfrowych liczb, suma cyfr których równa jest 20.

Przykład 1.28[18]**.** (Zadanie o nieporządkach.) Dane jest n różnych przedmiotów $a_1, \dots, a_n$ i n różnych komórek. Ile jest sposobów na to, by rozmieścić przedmioty po komórkach tak, żeby żaden przedmiot a_i nie trafił do komórki b_i.

Rozwiązanie. Jako zbiór X przyjmiemy zbiór wszystkich możliwych rozmieszczeń przedmiotów w komórkach. Wtedy $N = n!$. Wprowadzimy własności $\alpha_i =$ $=$(przedmiot a_i trafił do komórki b_i). Wtedy $N(\alpha_{i_1}, \dots, \alpha_{i_k}) = (n-k)!$,

$$S_k = \sum_{1 \leqslant i_1 < \dots < i_k \leqslant n} N(\alpha_{i_1}, \dots, \alpha_{i_k}) = C_n^k (n-k)! = \frac{n!}{k!}.$$

[16] 1.
[17] 36.
[18] $n! \sum_{k=0}^{n} (-1)^k 1/k!$.

Przykad 1.29. Ile jest liczb od 1 do 1000 niepodzielnych ani przez 2, ani przez 3, ani przez 5?

Rozwiązanie.

1. $X = \{1..1000\}$, $N = 1000$.
2. $\alpha_1(x) = (x \text{ jest podzielna przez } 2)$,
3. $\alpha_2(x) = (x \text{ jest podzielna przez } 3)$,
4. $\alpha_3(x) = (x \text{ jest podzielna przez } 5)$.
5. $N(\alpha_1) = 500$, $N(\alpha_2) = 333$, $N(\alpha_3) = 200$, $S_1 = 1033$.
6. $N(\alpha_1, \alpha_2) = 166$, $N(\alpha_1, \alpha_2) = 100$, $N(\alpha_2, \alpha_3) = 66$, $S_2 = 332$.
7. $N(\alpha_1, \alpha_2, \alpha_3) = 33$, $S_3 = 33$.
8. $N_0 = 1000 - 1033 + 332 - 33 = 266$.

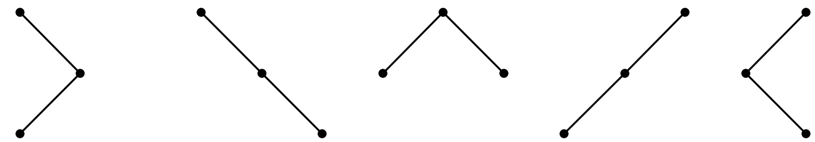
8. Pytania na egzamin

1. Niech X będzie zbiorem. Co oznacza się przez $|X|$?
2. Sformułować zasadę sumy.
3. Udowodnić zasadę sumy.
4. Sformułować zasadę iloczynu.
5. Udowodnić zasadę iloczynu.
6. Co nazywa się (n,r)-próbką?
7. Co nazywa się (n,r)-próbką uporządkowaną?
8. Co nazywa się (n,r)-próbką nieuporządkowaną?
9. Co nazywa się (n,r)-próbką z powtórzeniami?
10. Co nazywa się (n,r)-wariacją?
11. Co nazywa się (n,r)-kombinacją?
12. Podać przykład (n,r)-kombinacji bez powtórzeń.
13. Podać przykład (n,r)-kombinacji z powtórzeniami.
14. Podać przykład (n,r)-wariacji bez powtórzeń.
15. Podać przykład (n,r)-wariacji z powtórzeniami.
16. Co nazywa się permutacją?
17. Co oznacza się przez $\bar{A}_n^r$?
18. Co oznacza się przez A_n^r?
19. Co oznacza się przez $\bar{C}_n^r$?
20. Co oznacza się przez C_n^r?
21. Podać wzór na $\bar{C}_n^r$.
22. Podać wzór na $\bar{A}_n^r$.
23. Podać wzór na C_n^r.
24. Podać wzór na A_n^r.
25. Podać wzór na ilość permutacji.
26. Udowodnić wzór na $\bar{C}_n^r$.
27. Udowodnić wzór na $\bar{A}_n^r$.
28. Udowodnić wzór na C_n^r.

5. Zagadnienia z kombinatoryki, związane z liczbami Catalana

Na rysunkach 6.1 i 6.2 przedstawione są niektóre zagadnienia z kombinatoryki, związane z liczbami Catalana.

Definicja 6.11. *Drzewem binarnym* o n wierzchołkach nazywa się *graf pusty*, jeżeli $n = 0$, oraz przy $n > 1$ graf, mający wierzchołek k, nazywany *korzeniem*, *lewe poddrzewo* L o l wierzchołkach i *prawe poddrzewo* R o r wierzchołkach. Przy czym $l + r + 1 = n$.



Rysunek 6.1. Drzewa binarne o trzech wierzchołkach

6. Pytania na egzamin

1. Zdefiniować funkcję tworzącą ciągu.
2. Zdefiniować sumę funkcji tworzących.
3. Zdefiniować iloczyn funkcji tworzącej i liczby rzeczywistej.
4. Zdefiniować iloczyn funkcji tworzących.
5. Podać przykład funkcji tworzącej.
6. Co będzie funkcja tworzącą la ciągu $a_k = C_n^k$?
7. Co będzie funkcja tworzącą la ciągu $a_k = \bar{C}_n^k$?
8. Niech $X = \{a_1, a_2, a_3\}$, c_k będzie ilością k-kombinacji z powtórzeniami, wśród elementów których a_1 może występować co najwyżej dwa razy, a_2 — trzy, zaś a_3 — cztery razy. Jaka będzie funkcja tworząca dla ciągu c_k?
9. Niech $X = \{a_1, a_2, a_3\}$, c_k będzie ilością k-kombinacji z powtórzeniami, wśród elementów których a_1 może występować jeden lub dwa razy, a_2 — zero lub trzy razy, zaś a_3 — zero, dwa lub cztery razy. Jaka będzie funkcja tworząca dla ciągu c_k?
10. Niech $X = \{a_1, a_2, a_3\}$, c_k będzie ilością k-kombinacji z powtórzeniami, wśród elementów których a_1 może występować zero lub jeden raz, a_2 — zero lub trzy razy, zaś a_3 — dowolną ilość razy. Jaka będzie funkcja tworząca dla ciągu c_k?
11. Niech $X = \{a_1, a_2, a_3\}$, c_k będzie ilością k-kombinacji z powtórzeniami, wśród elementów których a_1 może występować zero lub jeden raz, a_2 — dowolną parzystą ilość razy, natomiast a_3 — dowolną nieparzystą ilość razy. Jaka będzie funkcja tworząca dla ciągu c_k?
12. Podać definicję liczb Fibonacciego.
13. Podać pierwsze pięć liczb Fibonacciego.
14. Wypisać funkcję tworzącą dla liczb Fibonacciego.

przy czym

$$\begin{cases} A\frac{1-\sqrt{5}}{2} + B\frac{1+\sqrt{5}}{2} = -1, \\ A + B = 0, \end{cases}$$

skąd

$$A = \frac{1}{\sqrt{5}}, \quad B = -\frac{1}{\sqrt{5}}.$$

Mamy zatem

$$\begin{aligned} F(x) &= \frac{1}{\sqrt{5}}\left(\frac{1}{1-\frac{1+\sqrt{5}}{2}x} - \frac{1}{1-\frac{1-\sqrt{5}}{2}x}\right) = \\ &= \frac{1}{\sqrt{5}}\left(\sum_{k=0}^{\infty}\left(\frac{1+\sqrt{5}}{2}\right)^k x^k - \sum_{k=0}^{\infty}\left(\frac{1-\sqrt{5}}{2}\right)^k x^k\right) = \\ &= \sum_{k=0}^{\infty}\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right]x^k. \end{aligned}$$

I ostatecznie

$$F_k = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right].$$

4. Liczby Catalana

Definicja 6.6.

$$c_0 = 1,$$

$$c_{k+1} = c_0c_k + c_1c_{k-1} + \cdots + c_{k-1}c1 + c_kc_0 = \sum_{i=0}^{k} c_ic_{k-i}, \quad k = 1, 2, 3, \ldots$$

Oto początek ciągu liczb Catalana: $1, 1, 2, 5, 14, 42, 132$.

Rozważmy funkcję tworzącą dla liczb Catalana:

$$C(x) = c_0 + c_1x + c_2x^2 + \ldots.$$

Lemat 6.7. *Funkcja $C(x)$ spełnia równanie $C(x) = xC(x)C(x) + 1$.*

Wniosek 6.8. *Funkcja tworząca dla liczb Catalana równa jest*

$$C(x) = \frac{1-\sqrt{1-4x}}{2x}.$$

Wniosek 6.9. *Dla liczb Catalana zachodzi równość*

$$c_n = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n+1}C_{2n}^n.$$

Wniosek 6.10. *Dla liczb Catalana zachodzi równość $c_{n+1} = \frac{4n+2}{n+2}c_n$.*

29. Udowodnić wzór na A_n^r.
30. Udowodnić wzór na ilość permutacji.
31. Niech X i Y będą zbiorami. Co oznacza się przez Y^X?
32. Niech X i Y będą zbiorami. Podać wzór na $\left|Y^X\right|$.
33. Niech X i Y będą zbiorami. Udowodnić wzór na $\left|Y^X\right|$.
34. Ile istnieje różnowartościowych odwzorowań $X \to Y$?
35. Udowodnić wzór na ilość różnowartościowych odwzorowań $X \to Y$?
36. Na ile sposobów można pomalować romb, podzielony na pięć części, sześcioma kolorami dopuszczając jednakowe kolory?
37. Na ile sposobów można pomalować romb, podzielony na pięć części, sześcioma kolorami, jeśli różne części maluje się na różne kolory?
38. Ile jest sposobów na wybranie pięciu kostek z pełnego zestawu 28 kostek domina?
39. Niech X będzie $2n$-elementowym zbiorem ($n > 1$). Czego jest więcej: n-wariacji czy n-kombinacji elementów X?
40. Wśród 20 sztuk towaru jest 15 sztuk standardowych. Na ile sposobów można wybrać 4 sztuki w taki sposób, żeby w próbce były dokładnie 2 sztuki standardowe?
41. Wśród 20 sztuk towaru jest 15 sztuk standardowych. Na ile sposobów można wybrać 4 sztuki w taki sposób, żeby w próbce były nie mniej niż 2 sztuki standardowe?
42. Zestaw liczy 20 monet nominału 1, 2 oraz 5 złotych. Ile istnieje zestawów?
43. Ile istnieje całkowitych rozwiązań układu $\begin{cases} x_1 + \cdots + x_k = r, \\ x_i \geqslant a_i, \quad i = 1, \ldots, k, \end{cases}$ gdzie $n \geqslant 1$, a_i są liczby całkowite?
44. Ile istnieje całkowitych rozwiązań układu $\begin{cases} x_1 + x_2 + x_3 = 10, \\ x_i \geqslant i, \quad i = 1, 2, 3 \end{cases}$?
45. Udowodnić wzór na ilość całkowitych rozwiązań układu $\begin{cases} x_1 + \cdots + x_k = r, \\ x_i \geqslant 0, \quad i = 1, \ldots, k, \end{cases}$ gdzie $n \geqslant 1$.
46. Co nazywa się rozbiciem zbioru?
47. Co oznacza się przez $C_n^{n_1,n_2,\ldots,n_k}$, gdzie $n_1 + \cdots + n_k = n$?
48. Co oznacza się przez $N(m_1, m_2, \ldots, m_n)$, gdzie $m_1 + 2m_2 + \cdots + nm_n = n$?
49. Podać wzór na $C_n^{n_1,n_2,\ldots,n_k}$, gdzie $n_1 + \cdots + n_k = n$.
50. Podać wzór na $N(m_1, m_2, \ldots, m_n)$, gdzie $m_1 + 2m_2 + \cdots + nm_n = n$.
51. Udowodnić wzór na $C_n^{n_1,n_2,\ldots,n_k}$, gdzie $n_1 + \cdots + n_k = n$.
52. Udowodnić wzór na $N(m_1, m_2, \ldots, m_n)$, gdzie $m_1 + 2m_2 + \cdots + nm_n = n$.
53. Niech $X = \{\, a_a, \ldots, a_k \,\}$. Ile istnieje n-wariacji z powtórzeniami, wśród elementów których znajduje się n_1 elementów a_1, n_2 elementów a_2, ... , n_k elementów a_k?
54. Niech $X = \{\, a_a, \ldots, a_k \,\}$. Udowodnić, że istnieje dokładnie $C_n^{n_1,n_2,\ldots,n_k}$ n-wariacji z powtórzeniami, wśród elementów których znajduje się n_1 elementów a_1, n_2 elementów a_2, ... , n_k elementów a_k?

55. Grupa studentów liczy 20 osób. W wyborach samorządowych na P. Nowaka głosowało 8 osób, przeciw niemu — 7 osób, pięć osob wstrzymały się od głosu. Na ile sposobów mogło odbyć się takie głosowanie?
56. Na ile sposobów można pomalować romb, rozbity na osiem części w cztery kolory tak, żeby w pierwszy kolor zostało pomalowano 3 części, w drugi — 1, w trzeci — 2, w czwarty — 2?
57. Na ile sposobów z grupy liczącej 20 osób można stworzyć 2 koalicji po 6 osób i cztery koalicje po 2 osoby?
58. Podać wzór na $(x_1 + \cdots + x_n)^k$.
59. Udowodnić wzór na $(x_1 + \cdots + x_n)^k$.
60. Ustalić współczynnik przy $x_1^3x_2^4x_3^3x_4^5$ w rozwinięciu $(x_1+x_2+x_3+x_4)^{15}$.
61. Sformułować zasadę włączeń i wyłączeń.
62. Objaśnić, dlaczego zachodzi wzór włączeń i wyłączeń dla $n = 3$.
63. Niech $X = \{0, 1, \ldots, 100\}$. Obliczyć ilość elementów zbioru X, nie posiadających żadnej z wymienonych własności: a) x jest liczbą parzystą, b) $x > 6$, c)$2 < x < 88$.
64. Wyznaczyć ilość liczb całkowitych od 1 do 1000, które są niepodzielne ani przez 2, ani przez 5, ani przez 7, ani przez 3.

II. Algorytmy kombinatorne

1. Niezmiennik pętli

Definicja 2.1. *Niezmiennikiem pętli* nazywa się logiczne wyrażenie, wartość którego się nie zmienia podczas wykonywania pętli

Zastosowania niezmienników oparte są na następującym twierdzeniu.

Twierdzenie 2.2. *Jeżeli w pętli WHILE (algorytm 2.1) przed pętlą niezmiennik N ma wartość* **prawda***, to po pętli niezmiennik N ma wartość* **prawda***, zaś warunek kontynuacji pętli W ma wartość* **fałsz***.*

{*Niezmiennik:* N}
while W **do**
  Działania
end while

Algorytm 2.1. Pętla WHILE

Projektowanie pętli ma następujące etapy:

1. Wybór niezmiennika pętli.
2. Dobór warunku pętli w zależności od niezmiennika.
3. Projektowanie działań wstępnych, gwarantujących że niezmiennik ma wartość **prawda** przed pętlą.
4. Projektowanie ciała pętli, ze względem niezmiennika.

W szczególności, jeżeli każdy z elementów zbioru $X = \{ a_1, a_2, \ldots a_n \}$ może się pojawić w kombinacji dowolną ilość razy, funkcja tworząca będzie równa

$$\sum_{k=0}^{\infty} \bar{C}_n^k x^k = \underbrace{(1 + x + x^2 + \ldots) \ldots (1 + x + x^2 + \ldots)}_{n \text{ razy}} =$$

$$= \frac{1}{1-x} \cdots \frac{1}{1-x} = \frac{1}{(1-x)^n}.$$

Różniczkując k razy, otrzymamy

$$\frac{d^k}{dx^k} \frac{1}{(1-x)^n} = n(n+1) \ldots (n+k-1) \frac{1}{(1-x)^{n+k}}.$$

Więc

$$\bar{C}_n^k = \frac{n(n+1) \ldots (n+k-1)}{k!} = C_{n+k-1}^k.$$

3. Liczby Fibonacciego

Definicja 6.5.

$$F_0 = 0, \quad F_1 = 1,$$
$$F_{k+1} = F_k + F_{k-1}, \quad k = 2, 3, \ldots.$$

Oto początek ciągu Fibonacciego: $0, 1, 1, 2, 3, 5, 8, 13, 21, 34$.

Funkcja tworząca $F(x)$ dla ciągu Fibonacciego spełnia równanie

$$F(x) = \sum_{k=0}^{\infty} F_k x^k = 0 + x + \sum_{k=2}^{\infty} (F_{k-2} + F_{k-1}) x^k =$$
$$= x + x^2 \sum_{k=2}^{\infty} F_{k-2} x^{k-2} + x \sum_{k=2}^{\infty} F_{k-1} x^{k-1} =$$
$$= x + x^2 F(x) + x F(x) = x + (x^2 + x) F(x),$$

skąd

$$F(x) = \frac{x}{1 - x - x^2}.$$

Żeby uzyskać jawny wzór na F_n, przedstawimy $F(x)$ jako sumę ułamków najprostszych. Ponieważ $(1 - x - x^2) = \left(1 - \frac{1+\sqrt{5}}{2} x\right)\left(1 - \frac{1-\sqrt{5}}{2} x\right)$, powinna zachodzić równość

$$\frac{x}{1 - x - x^2} = \frac{A}{1 - \frac{1+\sqrt{5}}{2} x} + \frac{B}{1 - \frac{1-\sqrt{5}}{2} x},$$

Przykłady 6.4.

1. $a_k = 1, \quad A(x) = \sum\limits_{k=0}^{\infty} x^k = \frac{1}{1-x}.$
2. $a_k = \frac{1}{k!}, \quad A(x) = \sum\limits_{k=0}^{\infty} \frac{1}{k!} x^k = e^x.$
3. $a_k = C_n^k, \quad A(x) = \sum\limits_{k=0}^{\infty} C_n^k x^k = \sum\limits_{k=0}^{n} C_n^k x^k = (1+x)^n.$
4. $a_k = 2^k, \quad A(x) = \sum\limits_{k=0}^{\infty} 2^k x^k = \sum\limits_{k=0}^{\infty} (2x)^k = \frac{1}{1-2x}.$

2. Funkcje tworzące i kombinacje

Rozważmy wzór

$$(1+x)^n = \underbrace{(1+x)(1+x)\dots(1+x)}_{n \text{ razy}} = \sum_{k=0}^{\infty} C_n^k x^k$$

Każdy czynnik $(1+x)$ można traktować jako odpowiadający pewnemu elementowi a_i zbioru $X = \{ a_1, \dots, a_n \}$ i jako reprezentujący dwie możliwości pojawienia się tego elementu w podzbiorze: zero razy (składnik $x^0 = 1$) i jeden raz (składnik $x^1 = x$). Każdy podzbiór X jest jednoznacznie określony przez definicję ilości pojawienia się w nim każdego elementu, i. e. przez wybór jednego ze składników iloczynu $(1+x)\dots(1+x)$. Określony w ten sposób każdy składnik da wkład do współczynnika przy x^k, gdzie k jest ilością elementów podzbioru. Więc ilość k-elementowych podzbiorów zbioru X jest równa C_n^k.

Rozumowanie to można przenieść na kombinacje z powtórzeniami. Niech $X = \{ a_1, a_2, a_3, a_4 \}$. Oznaczmy przez c_k ilość k-kombinacji z powtórzeniami, w których a_1 może występować co najwyżej dwa razy, a_2 — trzy, a_3 — jeden, zaś a_4 — cztery razy. Funkcja tworząca dla ciągu c_k jest

$$\begin{aligned} C(x) = \sum_{k=0}^{\infty} c_k x^k = \\ &= (1+x+x^2)(1+x+x^2+x^3)(1+x)(1+x+x^2+x^3+x^4) = \\ &= 1 + 4x + 9x^2 + 15x^3 + 20x^4 + 22x^5 + 20x^6 + 15x^7 + 9x^8 + 4x^9 + x^{10}. \end{aligned}$$

Dobierając i-ty czynnik można nakładać dowolne ograniczenia na ilość wchodzeń elementu a_i. Na przykład, jeżeli element a_i może występować 0, 3 lub 7 razy, czynnik ma postać $1 + x^3 + x^7$; jeżeli a_i może występować dowolną parzystą liczbę razy, czynnik jest równy $1 + x^2 + x^4 + \dots = \frac{1}{1-x^2}$.

Przykłady

Przykład 2.3. Znaleźć $\max\{ a_1, a_2, \dots, a_n \}$.

Rozwiązanie: patrz algorytm 2.2.

Dane: $n \geqslant 1$
Wyniki: $M = \max\{ a_1, a_2, \dots, a_n \}$
  $k \leftarrow 1$
  $M \leftarrow a_1$
  $\{$*Niezmiennik:* $M = \max\{ a_1, \dots, a_k \}\}$
  while $k \neq n$ **do**
    $k \leftarrow k + 1$
    if $a_k > M$ **then**
      $M \leftarrow a_k$
    end if
  end while

Algorytm 2.2. Obliczenie $\max\{ a_1, \dots, a_n \}$

Przykład 2.4. Obliczyć $\sum_{i=1}^{n} a_i$.

Rozwiązanie: patrz algorytm 2.3.

Dane: $n \geqslant 1$
Wyniki: $S = \sum_{i=1}^{n} a_i$
  $k \leftarrow 1$
  $S \leftarrow a_1$
  $\{$*Niezmiennik:* $S = \sum_{i=1}^{k} a_i\}$
  while $k \neq n$ **do**
    $k \leftarrow k + 1$
    $S \leftarrow S + a_k$
  end while

Algorytm 2.3. Obliczenie $\sum_{i=1}^{n} a_i$

Przykład 2.5. Obliczyć x^n.

Rozwiązanie: patrz algorytmy 2.4–2.6.

Uwaga 2.6. Rozwiązania 2.4 i 2.5 wymagają $O(n)$ działań, zaś rozwiązanie 2.6 tylko $O(\log_2 n)$ działań.

Przykład 2.7. Obliczyć $a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$.

Rozwiązanie: patrz algorytm 2.7 *(Schemat Gornera)*.

Dane: $n \geqslant 0$
Wyniki: $S = x^n$
  $k \leftarrow 0$
  $S \leftarrow 1$
  $\{$*Niezmiennik:* $S = x^k\}$
  while $k \neq n$ **do**
    $k \leftarrow k + 1$
    $S \leftarrow S \cdot x$
  end while

Algorytm 2.4. Obliczenie x^n

Dane: $n \geqslant 0$
Wyniki: $S = x^n$
  $k \leftarrow n$
  $S \leftarrow 1$
  $\{$*Niezmiennik:* $S = x^{n-k}\}$
  while $k \neq 0$ **do**
    $k \leftarrow k - 1$
    $S \leftarrow S \cdot x$
  end while

Algorytm 2.5. Obliczenie x^n

Dane: $n \geqslant 0$
Wyniki: $S = x^n$
  $k \leftarrow n$
  $S \leftarrow 1$
  $b \leftarrow x$
  $\{$*Niezmiennik:* $S \cdot b^k = x^n\}$
  while $k \neq 0$ **do**
    if k jest liczbą parzystą **then**
      $k \leftarrow k/2$
      $b \leftarrow b^2$
    else $\{$*k jest liczbą nieparzystą*$\}$
      $k \leftarrow k - 1$
      $S \leftarrow S \cdot b$
    end if
  end while

Algorytm 2.6. Obliczenie x^n

2. Wariacje z powtórzeniami

Zadanie 2.8. Wydrukować wszystkie ciągi długości k z liczb $1, \ldots, n$.

24. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich ciągów z n zer, jedynek i dwójek, w których żadna cyfra nie pojawia się dwa razy pod rząd. Zaimplementuj procedurę `JestZPrawej`.
25. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich ciągów z n zer i dwójek, w których żadna cyfra nie pojawia się trzy razy pod rząd. Narysuj drzewo poszukiwania dla $n = 5$.

VI. Funkcje tworzące

1. Definicje

Definicja 6.1. Niech $a_0, a_1, a_2, \ldots$ będzie ciągiem. *Funkcją tworzącą* nazywamy szereg formalny

$$A(x) = \sum_{k=0}^{\infty} a_k x^k. \tag{6.1}$$

Definicja 6.2. Niech $A(x) = \sum_{k=0}^{\infty} a_k x^k$ i $B(x) = \sum_{k=0}^{\infty} b_k x^k$ będą funkcje tworzące.

1. *Sumą (różnicą)* nazywamy szereg $A(x) \pm B(x) = \sum\limits_{k=0}^{\infty} (a_k \pm b_k) x^k$.
2. *Iloczynem* funkcji $A(x)$ i liczby rzeczywistej λ nazywamy szereg $\lambda A(x) = \sum\limits_{k=0}^{\infty} (\lambda a_k) x^k$.
3. *Iloczynem (iloczynem Cauchy'ego)* nazywamy szereg

$$A(x) \cdot B(x) = \sum_{k=0}^{\infty} c_k x^k,$$

gdzie

$$c_k = \sum_{i=0}^{k} a_i b_{k-i} = a_0 b_k + a_1 b_{k-1} + \cdots + a_k b_0.$$

Uwaga 6.3. Jeżeli szereg (6.1) jest zbieżnym w otoczeniu zera, to funkcji tworzącej odpowiada funkcja rzeczywista, określona w tym samym otoczeniu zera. W tym przypadku działaniom, określonym w definicji 6.2 odpowiadają działania nad funkcjami rzeczywistymi. Szereg (6.1) jest rozwinięciem funkcji w szereg Maclaurina. Więc

$$a_k = \frac{A^{(k)}(0)}{k!}.$$

7. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 liście po lewej stronie od W.
8. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 liście z góry od W.
9. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 wierzchołki po prawej stronie od W.
10. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 wierzchołki po lewej stronie od W.
11. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 wierzchołki z góry od W.
12. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 wierzchołki z doły od W.
13. Zaprojektuj modyfikację rozwiązania zadania o hetmanach szachowych, żeby wydrukowane zostało tylko jedno umieszczenie.
14. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich ciągów długości k z liczb $1, \dots, n$. Narysuj drzewo poszukiwań dla $k = 2$, $n = 3$.
15. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich ciągów długości k z liczb $1, \dots, n$. Zaimplementuj procedurę `JestZGóru`.
16. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich n-permutacji. Narysuj drzewo poszukiwań dla $n = 3$.
17. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich n-permutacji. Zaimplementuj procedurę `JestZPrawej`.
18. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich k-elementowych podzbiorów zbioru $\{1, \dots, n\}$. Narysuj drzewo poszukiwań dla $k = 2$, $n = 3$.
19. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich k-elementowych podzbiorów zbioru $\{1, \dots, n\}$. Zaimplementuj procedurę `WGórę`.
20. Niech dane będą tablica liczb $a_1, \dots, a_n$ i liczba s. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do rozwiązania zadania, czy można przedstawić s jako sumę niektórych a_i. Narysuj drzewo poszukiwań dla $a = (2, 3, -1)$, $s = 2$.
21. Niech dane będą tablica liczb $a_1, \dots, a_n$ i liczba s. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do rozwiązania zadania, czy można przedstawić s jako sumę niektórych a_i. Zaimplementuj procedurę `WPrawo`.
22. Niech dane będą tablica liczb $a_1, \dots, a_n$ i liczba s. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do rozwiązania zadania, czy można przedstawić s jako sumę niektórych a_i. Narysuj drzewo poszukiwań dla $a = (2, 3, -1)$, $s = 2$.
23. Zaprojektuj sposób zastosowania algorytmu obejścia drzewa do wydrukowania wszystkich ciągów z n zer, jedynek i dwójek, w których żadna cyfra nie pojawia się dwa razy pod rząd. Narysuj drzewo poszukiwania dla $n = 5$.

Dane: $n \geqslant 0$
Wyniki: $S = a_0 + \dots + a_n x^n$
  $k \leftarrow 0$
  $S \leftarrow a_n$
  $\{$*Niezmiennik:* $S = a_n x^k + a_{n-1} x^{k-1} + \dots + a_{n-k} x^0\}$
  while $k \neq n$ **do**
    $k \leftarrow k + 1$
    $S \leftarrow S \cdot x + a_{n-k}$
  end while

ALGORYTM 2.7. Schemat Gornera

Definicja 2.9 (Uporządkowanie *leksyko-graficzne***).** Ciąg $a = a_1, \dots, a_k$ jest *poprzednikiem* ciągu $b = b_1, \dots, b_l$, jeżeli istnieje liczba naturalna $m \geqslant 0$, taka że

1. $a_i = b_i$ dla $i = 1, \dots, m$,
2. $a_{m+1} < b_{m+1}$ lub a_{m+1} nie istnieje, zaś b_{m+1} istnieje ($k = m < l$).

Oznaczenie: $a \prec b$.

Rozwiązanie zadania 2.8. Wydrukujmy ciągi w porządku leksyko-graficznym. Pierwszym ciągiem będzie $\underbrace{1, \dots, 1}_{k \text{ razy}}$, ostatnim — $\underbrace{n, \dots, n}_{k \text{ razy}}$. Patrz algorytm 2.8.

Dane: $n, k \geqslant 1$
Wyniki: Wydrukować wszystkie ciągi długości k z liczb $1, \dots, n$.
  $\{$*l jest ostatnim ciągiem*$\}$
  $l_1 \leftarrow n, \dots, l_k \leftarrow n$
  $x_1 \leftarrow 1, \dots, x_k \leftarrow 1$
  wydrukować x
  $\{$*Niezmiennik: wydrukowane są wszystkie ciągi do x włącznie*$\}$
  while $x \neq l$ **do**
    $x \leftarrow$ następny ciąg
    wydrukować x
  end while

ALGORYTM 2.8. Wydrukowanie wszystkich ciągów długości k z liczb $1, \dots, n$

Przejście do następnego ciągu w algorytmie 2.8 odbywa się w sposób następujący:

1. znaleźć pierwszy z prawej wyraz, mniejszy od n,
2. powiększyć go o jeden,

Dane: x nie jest ostatnim ciągiem
Wyniki: x jest następnym ciągiem w porządku leksyko-graficznym

$p \leftarrow k$
{*Niezmiennik:* $x_{p+1} = \cdots = x_k = n$}
while $x_p = n$ **do**
  $p \leftarrow p - 1$
end while
$\{x_p < n,\ x_{p+1} = \cdots = x_k = n\}$
$x_p \leftarrow x_p + 1$
$i \leftarrow p$
{*Niezmiennik:* $x_{p+1} = \cdots = x_i = 1$}
while $i \neq n$ **do**
  $i \leftarrow i + 1$
  $x_i \leftarrow 1$
end while

Algorytm 2.9. Przejście do następnego ciągu w algorytmie 2.8

3. następne wyrazy ciągu przyrównać do jedynki.

Implementacja dana jest w algorytmie 2.9.

Uwaga 2.10. Jeżeli wyrazami ciągu są liczby od 0 do $n-1$, to przejściu do następnego ciągu odpowiada dodawanie jedynki przy podstawie numeracji n.

Zadanie 2.11. Wydrukować wszystkie podzbiory zbioru $\{1, \ldots k\}$.

Podpowiedź. Podzbiory znajdują się we wzajemnie-jednoznacznym odwzorowaniu z ciągami długości k z zer i jedynek.

3. Permutacje

Zadanie 2.12. Wydrukować wszystkie permutacje zbioru $\{1, \ldots, n\}$

Rozwiązanie. Wydrukujmy permutacje w porządku leksyko-graficznym. Pierwszą permutacją będzie $1, 2, \ldots, n$, zaś ostatnią — $n, \ldots, 2, 1$. Algorytm jest podobny do algorytmu 2.8. Przejście do następnej permutacji odbywa się w sposób następujący (v. algorytm 2.10):

1. Znaleźć największe k, takie że $x_k < x_{k+1}$ oraz $x_{k+1} > \cdots > x_n$.
2. Powiększyć x_k w najmniejszy możliwy sposób, i. e., zamienić x_k z najmniejszą spośród $x_{k+1}, \ldots, x_n$ liczbą, która jest większa od x_k.
3. Umieścić liczby $x_{k+1}, \ldots\ x_n$ w porządku wzrostu, żeby otrzymana permutacja była najmniejszą. Można skorzystać z tego, że ciąg $x_{k+1}, \ldots, x_n$ jest malejącym.

Dane: Robot jest w dopuszczalnym k-stanie
Wyniki: `JestZPrawej` = **prawda i** nowy numer pionownicy dla hetmana k zapisany jest w `New` **lub** `JestZPrawej` = **fałsz i** nie ma możliwości umieścić hetmana k w tej samej poziomnicy

$y \leftarrow k$
$x \leftarrow P_k + 1$
{*Niezmiennik:* `JestZPrawej` = **fałsz i** *nie można umieścić hetmana w pionownicach* $P_{k+1}, \ldots, x-1$ **lub** `JestZPrawej` = **prawda i** *można umieścić hetmana w pionownicy* x}
while $x \leqslant n$ **i** `JestZPrawej` = **fałsz do**
  if H_x = **fałsz i** U_{x-y} = **fałsz i** V_{x+y} = **fałsz then**
    `JestZPrawej` ← **prawda**
  else {*Pionownica lub jedna z przekątnych nie jest wolna*}
    $x \leftarrow x + 1$
  end if
end while
if `JestZPrawej` = **prawda then**
  `New` ← x
end if

Algorytm 5.10. Procedura `JestZPrawej` do zadania o hetmanach

Dane: Robot jest w dopuszczalnym k-stanie **i** można umieścić hetmana w pionownicy `New`
Wyniki: Robot jest w następnym dopuszczalnym k-stanie.

$x \leftarrow P_k$
$y \leftarrow k$
$H_x \leftarrow$ **fałsz**
$U_{x-y} \leftarrow$ **fałsz**
$V_{x+y} \leftarrow$ **fałsz**
$x \leftarrow$ `New`
$P_k \leftarrow x$
$H_x \leftarrow$ **prawda**
$U_{x-y} \leftarrow$ **prawda**
$V_{x+y} \leftarrow$ **prawda**

Algorytm 5.11. Procedura `WPrawo` do zadania o hetmanach

2. Co nazywa się liściem w drzewie poszukiwań?
3. Co nazywa się korzeniem w drzewie poszukiwań?
4. Podaj przykład drzewa poszukiwań, które ma 6 liści.
5. Podaj przykład drzewa poszukiwań, które ma 6 wierzchołków.
6. Podaj przykład drzewa poszukiwań i wierzchołka W, tak żeby istniały dokładnie 3 liście po prawej stronie od W.

Dane: Robot jest w dopuszczalnym k-stanie
Wyniki: `JestZGóru` = **prawda i** numer pionownicy dla kolejnego hetmana zapisany jest w `New` **lub**
`JestZGóru` = **fałsz i** nie ma możliwości umieścić kolejnego hetmana

if $k = n$ **then**
  `JestZGóru` ← **fałsz**
else $\{k < n\}$
  `JestZGóru` ← **fałsz**
  $y \leftarrow k + 1$
  $x \leftarrow 1$
  {*Niezmiennik:* `JestZGóru` = **fałsz i** *nie można umieścić hetmana w pionownicach* $1, \ldots, x-1$ **lub**
    `JestZGóru` = **prawda i** *można umieścić hetmana w pionownicy* x}
  while $x \leqslant n$ **i** `JestZGóru` = **fałsz do**
    if H_x = **fałsz i** U_{x-y} = **fałsz i** V_{x+y} = **fałsz then**
      `JestZGóru` ← **prawda**
    else {*Pionownica lub jedna z przekątnych nie jest wolna*}
      $x \leftarrow x + 1$
    end if
  end while
  if `JestZGóru` = **prawda then**
    `New` ← x
  end if
end if

Algorytm 5.8. Procedura `JestZGóru` do zadania o hetmanach

W procedurze `WPrawo` trzeba „zdjąć” hetmana z pozycji (P_k, k) i umieścić jego w pozycji $(\texttt{New}, k)$. Implementacja — algorytm 5.11.

W procedurze `Wydrukować` trzeba sprawdzić, czy zostały umieszczone wszystkie hetmany i, jeżeli tak, to wydrukować rozwiązanie. Implementacja dana jest w algorytmie 5.12.

Dane: Robot jest w liściu.
Wyniki: Jeżeli liść odpowiada rozwiązaniu, to ono jest wydrukowane

if $k = n$ **then**
  Wydrukować P.
end if

Algorytm 5.12. Procedura `Wydrukować` do zadania o hetmanach

2. Pytania na egzamin

1. Zdefiniuj drzewo poszukiwań.

Dane: x nie jest ostatnią permutacją
Wyniki: x jest następną permutacją w porządku leksyko-graficznym

$k \leftarrow n - 1$
{*Niezmiennik:* $x_{k+1} > \cdots > x_n$}
while $x_k > x_{k+1}$ **do**
  $k \leftarrow k - 1$
end while
$\{x_k < x_{k+1}$ *oraz* $x_{k+1} > \cdots > x_n\}$
$t \leftarrow k + 1$
{*Niezmiennik:* $t \leqslant n$ *oraz* $x_{k+1} > \cdots > x_t > x_k$}
while $t \neq n$ i $x_{t+1} > x_k$ **do**
  $t \leftarrow t + 1$
end while
$\{x_{k+1} > \cdots > x_t > x_k > x_{t+1} > \cdots > x_n\}$
$x_k \leftrightarrows x_t$
$\{x_{k+1} > \cdots > x_n\}$
Przestawić $x_{k+1}, \ldots x_n$ w porządku odwrotnym.

Algorytm 2.10. Przejście do następnego ciągu w algorytmie wyliczenia permutacji

4. Podzbiory

Zadanie 2.13. Wydrukować wszystkie k-elementowe podzbiory zbioru $\{1, \ldots, n\}$.

Pierwsze rozwiązanie. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako ciąg zer i jedynek długości n, zawierających dokładnie k jedynek. Wydrukujemy je w porządku leksyko-graficznym. Pierwszym ciągiem będzie $\underbrace{0, \ldots, 0}_{n-k \text{ razy}}, \underbrace{1, \ldots, 1}_{k \text{ razy}}$. Ostatnim zaś — $\underbrace{1, \ldots, 1}_{k \text{ razy}}, \underbrace{0, \ldots, 0}_{n-k \text{ razy}}$.

Algorytm jest podobny do algorytmu 2.8. Przy przejściu do kolejnego ciągu x_s można powiększyć, tylko gdy $x_s = 0$, a po prawej stronie są jedynki. Przejście do następnego ciągu, przedstawiającego podzbiór odbywa się w sposób następujący (v. algorytm 2.11):

1. Znaleźć największe takie s, że $x_s = 0$, $x_{s+1} = 1$.
2. Zamienić x_s na jedynkę.
3. Rozstawić $x_{s+1}, \ldots, x_n$ tak, żeby na początku były same zera, a potem — same jedynki, i. e., żeby otrzymany ciąg był najmniejszym z możliwych.

Drugie rozwiązanie. Przedstawmy podzbiór poprzez przeliczenie jego elementów. Żeby każdy podzbiór został wydrukowany jeden raz, wymieniamy elementy w porządku rosnącym. W taki sposób trzeba wydrukować wszystkie rosnące ciągi długości k z liczb $1, \ldots, n$. Wydrukujmy je w porządku

Dane: x przedstawia nie ostatni podzbiór
Wyniki: x przedstawia następny podzbiór w porządku leksyko-graficznym

$s \leftarrow n - 1$
while $x_s \neq 0$ lub $x_{s+1} \neq 1$ **do**
  $s \leftarrow s - 1$
end while
$\{x_s = 0$ *oraz* $x_{s+1} = 1\}$
$x_s \leftarrow 1$
$I \leftarrow 0$
$k \leftarrow s$
$\{$*Niezmiennik: I zgadza się z ilością jedynek spośród* $x_{s+1}, \ldots, x_k\}$
while $k \neq n$ **do**
  $k \leftarrow k + 1$
  $I \leftarrow I + x_k$
end while
$x_{s+1} \leftarrow 0, \ldots, x_{n-I+1} \leftarrow 0$
$x_{n-I+2} \leftarrow 1, \ldots, x_n \leftarrow 1$

Algorytm 2.11. Przejście do następnego ciągu w pierwszym algorytmie wyliczania podzbiorów długości k zbioru $\{1, \ldots, n\}$

leksyko-graficznym. Pierwszym ciągiem będzie $1, 2, \ldots, k$, zaś ostatnim — $(n - k + 1), \ldots, (n - 1), n$.

Przy przejściu do kolejnego ciągu x_s można powiększyć, tylko gdy $x_s < n - k + s$, więc przejście do kolejnego ciągu odbywa się w sposób następujący (v. algorytm 2.12):

1. Znaleźć największe takie s, że $x_s < n - k + s$.
2. Powiększyć x_s o jedynkę.
3. Kolejne wyrazu ciągu powinny rosnąć o jeden.

5. Pytania na egzamin

1. Co nazywa się niezmiennikiem pętli?
2. Niech przed pętlą WHILE niezmiennik pętli będzie mieć wartość **prawda**. Jakie wartości będą miały po pętle niezmiennik oraz warunek kontynuacji pętli?
3. Podaj przykład pętli WHILE.
4. Zaprojektuj pętlę do odnalezienia $\max\{y_1, \ldots, y_t\}$ używając niezmiennik.
5. Zaprojektuj pętlę do obliczania iloczynu $y_1 \cdot y_2 \ldots y_t$ używając niezmiennik.
6. Zaprojektuj pętlę do obliczania sumy $y_1 - y_2 + y_3 - \cdots + (-1)^{t-1} y_t$ używając niezmiennik.
7. *Liczba zmiennoprzecinkowa* przedstawiona jest jako para (x, k), gdzie $x \in [1, 10)$ jest *mantysą*, $k \in \mathbb{Z}$ nazywa się *wyznacznikiem części potęgowej*.

Wyniki: `JestZDołu` = **fałsz** $\iff$ Robot jest w korzeniu

if $k = 0$ **then**
  `JestZDołu` $\leftarrow$ **fałsz**
else $\{k > 0\}$
  `JestZDołu` $\leftarrow$ **prawda**
end if

Algorytm 5.6. Procedura `JestZDołu` do zadania o hetmanach

Dane: Robot jest w dopuszczalnym stanienie, nie w korzeniu $(k > 0)$
Wyniki: Robot jest w dopuszczalnym stanie o jeden poziom niżej

$x \leftarrow P_k$; $y \leftarrow k$
$H_x \leftarrow$ **fałsz**
$U_{x-y} \leftarrow$ **fałsz**
$V_{x+y} \leftarrow$ **fałsz**
$k \leftarrow k - 1$

Algorytm 5.7. Procedura `WDół` do zadania o hetmanach

`JestZGóru` powinno być **fałsz**. Jeżeli $k < n$, to trzeba dobrać taką wartość P_{k+1} między 1 a n, żeby odpowiedne pionownica i dwie przekątne były wolne. Jeżeli takiej wartości nie ma, `JestZGóru` powinno być **fałsz**. Jeżeli `JestZGóru` = **fałsz**, wartość P_{k+1} nie jest istotną. Implementacja procedury dana jest w algorytmie 5.8.

W procedurze `WGórę` trzeba umieścić kolejnego hetmana w pionownicy P_{k+1}, i. e., oznaczyć pionownicę i obydwie przekątne, jako zajęte i powiększyć poziom Robota w drzewie. Implementacja procedury dana jest w algorytmie 5.9.

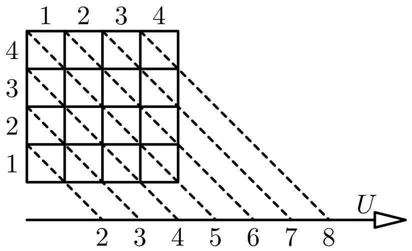
Dane: Robot jest w dopuszczalnym k-stanie **i** można umieścić kolejnego hetmana w pinownicy `New`
Wyniki: Robot jest w pierwszym dopuszczalnym $(k + 1)$-stanie.

$k \leftarrow k + 1$
$x \leftarrow$ `New`
$y \leftarrow k$
$P_k \leftarrow x$
$H_x \leftarrow$ **prawda**
$U_{x-y} \leftarrow$ **prawda**
$V_{x+y} \leftarrow$ **prawda**

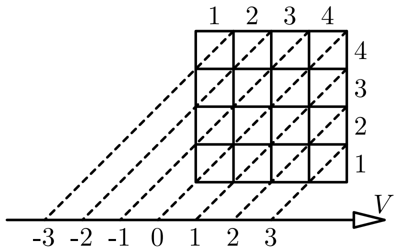
Algorytm 5.9. Procedura `WGórę` do zadania o hetmanach

Procedura `JestZPrawej` jest podobna do procedury `JestZGóru`. Różnica polega na tym, że dobieramy pionownicę dla hetmana w tej samej poziomnicy. Implementacja procedury dana jest w algorytmie 5.10.

poziomnicy y i pionownicy x, znajduje się na prawej przekątnej $x+y$, v. rysunek 5.9. V_i ma wartość **prawda**, jeżeli na i-ej *lewej przekątnej* znajduje się hetman, odpowiednio **fałsz**, jeżeli i-ta lewa przekątna jest wolna. Hetman, umieszczony na poziomnicy y i pionownicy x, znajduje się na lewej przekątnej $x-y$, v. rysunek 5.10.



Rysunek 5.9. Prawe przekątne



Rysunek 5.10. Lewe przekątne

Tablicy H, U i V pomogą sprawdzić dopuszczalność stanu. Na przykład, dla stanu $\mathcal{A}$ na rysunku 5.8 tablicy te równe są odpowiednio

$$H = (\textbf{prawda}, \textbf{prawda}, \textbf{fałsz}, \textbf{fałsz}),$$
$$U = (\textbf{prawda}, \textbf{fałsz}, \textbf{fałsz}, \textbf{prawda}, \textbf{prawda}, \textbf{fałsz}, \textbf{fałsz}),$$
$$V = (\textbf{fałsz}, \textbf{fałsz}, \textbf{prawda}, \textbf{prawda}, \textbf{fałsz}, \textbf{prawda}, \textbf{fałsz}).$$

W korzeniu wszystkie wartości tych tablic równe są **fałsz**.

Do rozwiązania zadania zastosujemy algorytm 5.2. Główny program, łącznie z działaniami wstępnymi, przedstawiony jest w algorytmie 5.5.

Dane: $n > 0$
Wyniki: wydrukowane są wszystkie rozwiązania zadania o hetmanach
  $k \leftarrow 0$
  $U_2 \leftarrow$ **fałsz**, … , $U_{2n} \leftarrow$ **fałsz**
  $V_{1-n} \leftarrow$ **fałsz**, … , $V_{n-1} \leftarrow$ **fałsz**
  $H_1 \leftarrow$ **fałsz**, … , $H_n \leftarrow$ **fałsz**
  `algorytm 5.2`

Algorytm 5.5. Program główny do zadania o hetmanach

Do implementacja procedur `JestZDołu` oraz `WDół` zauważmy, że poziom Robota na drzewie określony jest poprzez wartość zmiennej k (ilości umieszczonych hetmanów). W szczególności, w korzeniu $k = 0$. Implementacja procedur dana jest w algorytmach 5.6 i 5.7.

W procedurze `JestZGóru` trzeba dobrać pionownicę dla kolejnego hetmana. W szczególności, jeżeli $k = n$, i.e. wszystkie hetmany zostały umieszczone,

Dane: x przedstawia nie ostatni podzbiór
Wyniki: x przedstawia następny podzbiór w porządku leksyko-graficznym
  $s \leftarrow n$
  while $x_s \geqslant n-k+s$ **do**
    $s \leftarrow s-1$
  end while
  $x_s \leftarrow x_s+1$
  $i \leftarrow s$
  {*Niezmiennik: w ciągu $x_s, \ldots, x_i$ każdy następny wyraz jest o 1 większy od poprzedniego*}
  while $i \neq n$ **do**
    $i \leftarrow i+1$
    $x_i \leftarrow x_{i-1}+1$
  end while

Algorytm 2.12. Przejście do następnego ciągu w drugim algorytmie wyliczania podzbiorów długości k zbioru $\{1, \ldots, n\}$

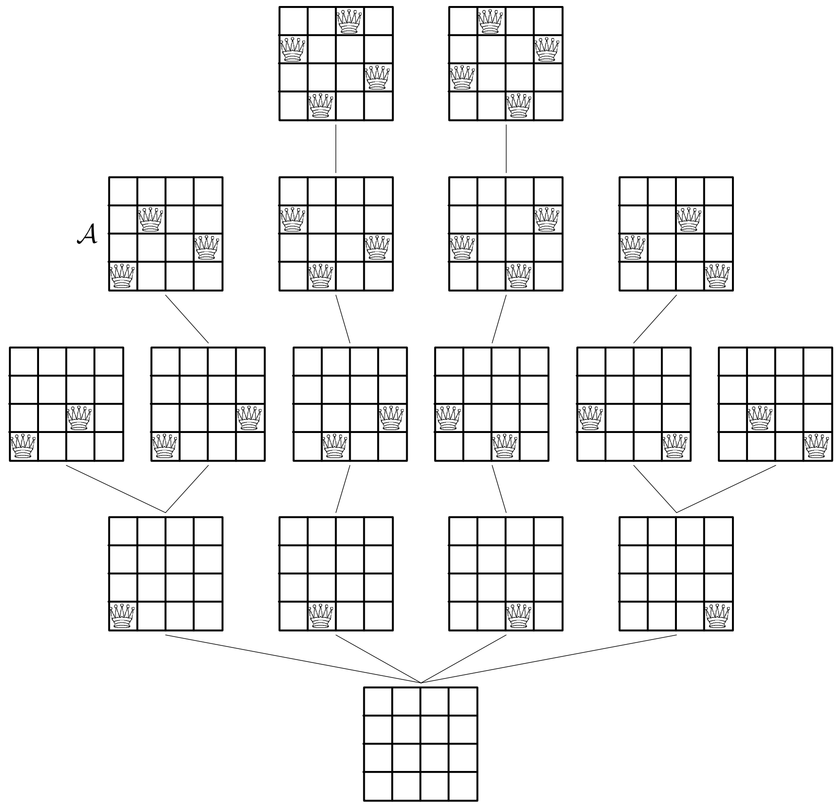
Wartość liczby zmiennoprzecinkowej to $x \cdot 10^k$. Opracować algorytm do podniesienia liczby zmiennoprzecinkowej (x,k) do potęgi s o skomplikowaności $O(s)$.

8. *Liczba zmiennoprzecinkowa* przedstawiona jest jako para (x,k), gdzie $x \in [1,10)$ jest *mantysą,* $k \in \mathbb{Z}$ nazywa się *wyznacznikiem części potęgowej*. Wartość liczby zmiennoprzecinkowej to $x \cdot 10^k$. Opracować algorytm do podniesienia liczby zmiennoprzecinkowej (x,k) do potęgi s o skomplikowaności $O(\log_2 s)$. Podpowiedź: można użyć niezmiennik $(y \cdot 10^\alpha) \cdot (b \cdot 10^\beta)^\lambda = (x \cdot 10^k)^s$, $1 \leqslant y, b < 10$.
9. Obliczyć według schematu Gornera wartość wielomianu $x^5 - 2x^4 + 3x^3 - 4x^2 + 5$ w punkcie 2.
10. Uporządkować w porządku leksyko-graficznym ciągi: $a_1 = (1,2,3)$, $a_2 = (1,1,1)$, $a_3 = (1,1)$, $a_4 = (1)$.
11. Podaj definicję porządku leksyko-graficznego.
12. Podaj przykład ciągu, który jest między $c_1 = (1,1,1)$ a $c_2 = (1,1)$ względem porządku leksyko-graficznego.
13. Podaj przykład ciągu x, takiego, że $x \prec y$, gdzie $y = (0,0,0)$.
14. Ustal wzajemnie-jednoznaczne odwzorowanie zbioru wszystkich podzbiorów zbioru $\{1, 2, \ldots, n\}$ na zbiór wszystkich ciągów długości n, złożonych z zer i jedynek. Co odpowiada zbiorowi pustemu?
15. Zaprojektuj algorytm wydrukowania wszystkich ciągów długości k z liczb $1, \ldots, n$ w porządku, odwrotnym do leksyko-graficznego. (Bez implementacji przejścia do następnego ciągu).
16. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich ciągów długości k z liczb $1, \ldots, n$ w porządku, odwrotnym do

leksyko-graficznego.

17. Zaprojektuj algorytm wydrukowania wszystkich podzbiorów zbioru $\{1, 2, \ldots, k\}$.
18. Zaprojektuj algorytm wydrukowania wszystkich permutacji zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego. (Bez implementacji przejścia do następnego ciągu).
19. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich permutacji zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego.
20. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako ciąg zer i jedynek długości n, zawierających dokładnie k jedynek. Zaprojektuj algorytm wydrukowania wszystkich k-elementowych podzbiorów zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego. (Bez implementacji przejścia do następnego ciągu).
21. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako ciąg zer i jedynek długości n, zawierających dokładnie k jedynek. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich k-elementowych podzbiorów zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego.
22. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako rosnący ciąg jego elementów. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich k-elementowych podzbiorów zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego.
23. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako malejący ciąg jego elementów. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich k-elementowych podzbiorów zbioru $\{1, \ldots, n\}$ w porządku, odwrotnym do leksyko-graficznego.
24. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako malejący ciąg jego elementów. Zaprojektuj algorytm przejścia do kolejnego ciągu przy wydrukowaniu wszystkich k-elementowych podzbiorów zbioru $\{1, \ldots, n\}$ w porządku leksyko-graficznym.
25. Wypisać wszystkie 3-elementowe ciągi z liczb $\{0, 1\}$ w porządku leksyko-graficznym.
26. Wypisać wszystkie 2-elementowe ciągi z liczb $\{0, 1, 2\}$ w porządku, odwrotnym do leksyko-graficznego.
27. Wypisać wszystkie podzbiory zbioru $\{0, 1\}$ w porządku leksyko-graficznym.
28. Wypisać wszystkie podzbiory zbioru $\{0, 1, 2\}$ w porządku, odwrotnym do leksyko-graficznego.
29. Wypisać wszystkie permutacje zbioru $\{0, 1\}$ w porządku leksyko-graficznym.
30. Wypisać wszystkie permutacje zbioru $\{0, 1, 2\}$ w porządku, odwrotnym do leksyko-graficznego.
31. Przedstawmy k-elementowy podzbiór zbioru $\{1, \ldots, n\}$ jako malejący ciąg jego elementów. Wypisać wszystkie 2-elementowe podzbiory zbioru

sensu kontynuować poszukiwanie po tej gałęzi. Nazwiemy k-stan *dopuszczalnym,* jeżeli hetmany się nie atakują. Rozważmy drzewo, złożone tylko z *dopuszczanych* stanów (v. rysunek 5.8).



RYSUNEK 5.8. Drzewo stanów dopuszczalnych w zadaniu o hetmanach szachowych dla $n = 4$

Przedstawmy stan za pomocą zmiennej k, ilości ustawionych hetmanów, oraz tablicy P_i, $i = 1, \ldots n$, gdzie P_i jest numerem pionownicy hetmana numer i, ustanowionego w poziomnicy i. Na przykład, stanowi $\mathcal{A}$ na rysunku 5.8 odpowiada $k = 3$, $P = (1, 4, 2, *)$, w korzeniu $k = 0$. Przy $i > k$ wartość P_i nie jest istotną.

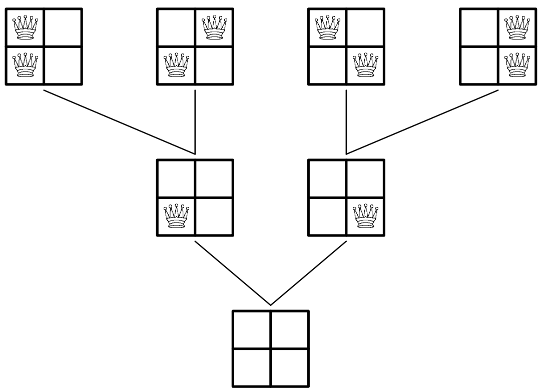
Wprowadzimy trzy tablicy pomocnicze H_i, $i = 1, \ldots, n$, U_i, $i = 2, \ldots 2n$ oraz V_j, $j = 1 - n, \ldots, n - 1$. H_i ma wartość **prawda**, jeżeli na i-ej pionownicy znajduje się hetman, oraz **fałsz**, jeżeli i-ta pionownica jest wolna. U_i ma wartość **prawda**, jeżeli na i-ej *prawej przekątnej* znajduje się hetman, oraz **fałsz**, jeżeli i-ta prawa przekątna jest wolna. Hetman, umieszczony na

Dane: Robot jest w korzeniu, wierzchołki nie wydrukowane
Wyniki: Robot jest w korzeniu, wierzchołki są wydrukowane

```
{Wydrukowane są wszystkie wierzchołki po lewej stronie i z dołu}
WGóręIWydrukować
{Niezmiennik: Wydrukowane są wszystkie wierzchołki po lewej stronie,
  z dołu i z góry}
while JestZDołu do
  if JestZPrawej then
    WPrawo
    {Wydrukowane są wszystkie wierzchołki po lewej stronie i z dołu}
    WGóręIWydrukować
  else {Nie ma z prawa, jest z dołu}
    WDół
  end if
end while
{Robot jest w korzeniu, więc wydrukowane są wszystkie wierzchołki}
```

Algorytm 5.4. Wydrukowanie wszystkich wierzchołków drzewa poszukiwań

$(k+1)$-ty hetman. Otrzymany graf będzie drzewem poszukiwań i można zastosować program obejścia drzewa 5.2, wybierając do druku tylko te liście, w których hetmany się nie atakują.



Rysunek 5.7. Drzewo stanów w zadaniu o hetmanach szachowych dla $n = 2$

Drzewo poszukiwania, określone w poprzednim akapicie, ma n^n liści i $\frac{n^{n+1}-1}{n-1}$ wierzchołków. W taki sposób, można spodziewać się algorytmu o skomplikowaności $O(n^n)$. Można powiększyć efektywność algorytmu, zmniejszywszy drzewo. Zauważmy, że jeżeli w k-stanie dwa hetmany się atakują, to nie ma

$\{0,1,2,3\}$ w porządku leksyko-graficznym.

32. Przedstawmy k-elementowy podzbiór zbioru $\{1,\dots,n\}$ jako rosnący ciąg jego elementów. Wypisać wszystkie 3-elementowe podzbiory zbioru $\{0,1,2,3\}$ w porządku, odwrotnym do leksyko-graficznego.
33. Przedstawmy k-elementowy podzbiór zbioru $\{1,\dots,n\}$ jako ciąg z zer i jedynek, zawierający dokładnie k jedynek. Wypisać wszystkie 3-elementowe podzbiory zbioru $\{0,1,2,3\}$ w porządku, odwrotnym do leksyko-graficznego.
34. Przedstawmy k-elementowy podzbiór zbioru $\{1,\dots,n\}$ jako ciąg z zer i jedynek, zawierający dokładnie k jedynek. Wypisać wszystkie 2-elementowe podzbiory zbioru $\{0,1,2,3\}$ w porządku leksyko-graficznym.

III. Podzielność liczb naturalnych

1. Dzielenie liczb całkowitych i pierścień $\mathbb{Z}_m$

Twierdzenie 3.1. Możliwość dzielenia z resztą. $\forall x, y \in \mathbb{Z}$, $y \neq 0$ *istnieją jedyne* $q, r \in \mathbb{Z}$, $0 \leqslant r \leqslant |y| - 1$, *takie że*

$$x = qy + r,$$

$q = x \operatorname{div} y$ *nazywa się ilorazem,* $r = x \bmod y$ — *resztą.*

Definicja 3.2. Liczba całkowita x jest *podzielna* przez y (*wielokrotnością* y), jeżeli reszta przy dzieleniu wynosi zero, czyli $\exists q \in \mathbb{Z}$, $x = qy$. Mówimy również, że y jest *dzielnikiem (podzielnikiem)* x, piszemy $y|x$.

Ustalmy liczbę $m \in \mathbb{Z}$, $m > 1$.

Definicja 3.3. Liczby całkowite x i y są *porównywalne modulo* m, jeżeli $x - y$ jest podzielna przez m:

$$x \equiv y \pmod{m}.$$

Lemat. $x \equiv y \pmod{m} \iff x$ *i* y *mają jednakowe reszty przy dzieleniu przez* m

Twierdzenie 3.4. *Relacja porównywalności jest relacją równoważności.*

Dowód. Sprawdźmy, że relacja ta jest *zwrotną, symetryczną*, oraz *przechodnią.*

Wniosek 3.5. *relacją porównywalności rozbija zbiór* $\mathbb{Z}$ *na klasy reszt* $[x] = \{y \in \mathbb{Z} \mid y \equiv x \pmod{m}\}$. *Przy tym*

1. $[x_1] = [x_2] \iff x_1 \equiv x_2 \pmod{m}$,
2. *jeżeli* $x_1 \not\equiv x_2 \pmod{m}$, *to* $[x_1] \cap [x_2] = \emptyset$.

Przykład 3.6. Niech $m = 5$. Wtedy

1. $[1] = [-4] = [6] = \{\ldots, -9, -4, 1, 6, 11, \ldots\}$,
2. $[2] = [7] = \{\ldots, -3, 2, 7, \ldots\}$, $[1] \cap [2] = \emptyset$.

Twierdzenie 3.7. *Jeżeli* $x_1 \equiv y_1 \pmod{m}$ *oraz* $x_2 \equiv y_2 \pmod{m}$, *to* $x_1 \pm \pm x_2 \equiv y_1 \pm y_2 \pmod{m}$, *oraz* $x_1 x_2 \equiv y_1 y_2 \pmod{m}$

Definicja 3.8 (Dodawanie i mnożenie klas reszt).

1. $[x] + [y] = [x + y]$,
2. $[x] \cdot [y] = [xy]$.

Przykład 3.9. Niech $m = 5$. Wtedy $[2] \cdot [-2] = [-4] = [1]$, $[2] \cdot [7] = [14] = [4] = [-1]$.

Twierdzenie 3.10. *Określone w 3.8 działania dodawania i mnożenia zadowalają*

1. $[x] + [y] = [y] + [x]$,
2. $[x] + ([y] + [z]) = ([x] + [y]) + [z]$,
3. $[0] + [x] = [x] + [0] = [x]$,
4. $[x] + [-x] = [0]$,
5. $[x] \cdot [y] = [y] \cdot [x]$,
6. $[x] \cdot ([y] \cdot [z]) = ([x] \cdot [y]) \cdot [z]$,
7. $[1] \cdot [x] = [x] \cdot [1] = [x]$,
8. $[x] \cdot ([y] + [z]) = [x] \cdot [y] + [x] \cdot [z]$,
9. $([x] + [y]) \cdot [z] = [x] \cdot [z] + [y] \cdot [z]$.

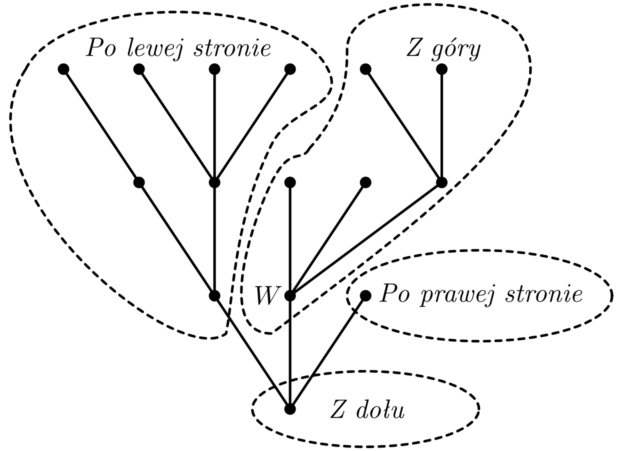
Definicja 3.11. Zbiór klas reszt razem z określonymi w 3.8 działaniami mnożenia i dodawania nazywa się *pierścieniem* $\mathbb{Z}_m$.

2. Kryteria podzielności

Twierdzenie 3.12. Kryterium podzielności przez 3 i 9. *Liczba całkowita x jest podzielna przez 3 (9) $\iff$ suma cyfr x jest podzielna przez 3 (9).*

Dowód. Rozważymy zapis dziesiętny liczby $x = a_n \ldots a_1 a_0$, gdzie a_i są cyfry. Wtedy

$$x = a_0 + 10a_1 + \cdots + 10^n a_n = (a_0 + \cdots + a_n) + 9a_1 + 99a_2 + \cdots + \underbrace{9\ldots9}_{n \text{ razy}} a_n \equiv$$
$$\equiv a_0 + \cdots + a_n \pmod{3(9)}$$



Rysunek 5.6. Klasy wierzchołków

Dane: Wydrukowane są wszystkie wierzchołki *po lewej stronie* i *z dołu*
Wyniki: Wydrukowane są wszystkie wierzchołki *po lewej stronie*, *z dołu* i *z góry*

{*Niezmiennik: Wydrukowane są wszystkie wierzchołki* po lewej stronie i z dołu}
while `JestZGóru` **do**
`Wydrukować`
`WGórę`
end while
{*Wydrukowane są wszystkie wierzchołki* po lewej stronie i z dołu, *Robot jest w liściu*}
`Wydrukować`
{*Wydrukowane są wszystkie wierzchołki* po lewej stronie, z dołu *i* z góry}

Algorytm 5.3. Modyfikowana procedura `WGóręIWydrukować`

Uwaga 5.4. Ilość działań w algorytmie jest $O(n)$, gdzie n jest ilością wierzchołków.

Zadanie 5.5. Na szachownicy $n \times n$ ustawić n hetmanów w taki sposób, żeby żadne dwa hetmany nie atakowały się. Wydrukować wszystkie ustawienia.

Rozwiązanie. Na każdej z n poziomnic powinien znajdować się jeden hetman. Nazwiemy k-stanem dowolne ustawienie k hetmanów na dolnych k poziomnicach. Określmy *drzewo stanów* (v. rysunek 5.7). Korzeniem będzie *stan pusty.* Z każdego k-stanu wychodzi n krawędzi do $k+1$ stanów, odpowiadającym umieszczeniu hetmana numer $k+1$ na poziomnicy $k+1$. Uporządkujmy $(k+1)$-stany według numeru *pionownicy,* w której został umieszczony

Dane: Wydrukowane są wszystkie liście *po lewej stronie*
Wyniki: Wydrukowane są wszystkie liście *po lewej stronie* i *z góry*
{*Niezmiennik: Wydrukowane są wszystkie liście* po lewej stronie}
while `JestZGóru` **do**
`WGórę`
end while
{*Wydrukowane są wszystkie liście* po lewej stronie, *Robot jest w liściu*}
`Wydrukować`

ALGORYTM 5.1. Procedura `WGóręIWydrukować`

Dane: Robot jest w korzeniu, liście nie wydrukowane
Wyniki: Robot jest w korzeniu, liście są wydrukowane
{*Wydrukowane są wszystkie liście* po lewej stronie}
`WGóręIWydrukować`
{*Niezmiennik: Wydrukowane są wszystkie liście* po lewej stronie *i* z góry}
while `JestZDołu` **do**
if `JestZPrawej` **then**
`WPrawo`
{*Wydrukowane są wszystkie liście* po lewej stronie}
`WGóręIWydrukować`
else {*Nie ma z prawej, jest z dołu*}
`WDół`
end if
end while
{*Robot jest w korzeniu, więc wydrukowane są wszystkie liście*}

ALGORYTM 5.2. Wydrukowanie wszystkich liści drzewa poszukiwań

Zadanie 5.3. Wydrukować *wszystkie* wierzchołki drzewa poszukiwań dokładnie jeden raz.

Rozwiązanie. Niech Robot będzie w pewnym wierzchołku W. Podzielmy wszystkie wierzchołki na klasy (rysunek 5.6):

1. *Z dołu* — ścieżka z korzenia do wierzchołka W przechodzi przez dany wierzchołek.
2. *Po lewej stronie* — ścieżka z korzenia do wierzchołka skręca w lewo przed wierzchołkiem W.
3. *Po prawej stronie* — ścieżka z korzenia do wierzchołka skręca w prawo przed wierzchołkiem W.
4. *Z góry* — ścieżka z korzenia do wierzchołka idzie przez wierzchołek W.

Modyfikacja procedury pomocniczej (cf. 5.1) dana jest w algorytmie 5.3.

Program główny podobny jest do algorytmu 5.2 i podany jest w algorytmie 5.4.

Twierdzenie 3.13. Kryterium podzielności przez 11. *Liczba całkowita x jest podzielna przez 11 $\iff$ różnica sumy parzystych i nieparzystych cyfr x jest podzielna przez 11.*

Dowód. Rozważymy zapis dziesiętny liczby $x = a_n \ldots a_1 a_0$, gdzie a_i są cyfry. Wtedy

$$\begin{aligned} x &= a_0 + 10a_1 + \cdots + 10^n a_n = a_0 + 11a_1 - a_1 + 99a_2 + a_2 + 1001a_3 - a_3 + \cdots = \\ &= (a_0 - a_1 + a_2 - a_3 + a_4 + \ldots) + (11a_1 + 99a_2 + 1\,001a_3 + 9\,999a_4 + \ldots) \equiv \\ &\equiv a_0 - a_1 + a_2 - a_3 + a_4 + \ldots \pmod{11}. \end{aligned}$$

Twierdzenie 3.14. Kryterium podzielności przez 7, 11, 13. *Liczba całkowita x jest podzielna przez 7 (11 albo 13) $\iff$ podzielna przez 7 (11 albo 13 odpowiednio) jest liczba następująca:*

Rozbijamy zapis dziesiętny liczby x z prawej strony na grupy po trzy cyfry. Otrzymamy ciąg liczb trzycyfrowych: T_0, $T_1, \ldots$. Określoną liczbą jest $T = T_0 - T_1 + T_2 - \ldots$.

Dowód. Rozważymy zapis dziesiętny liczby $x = a_n \ldots a_1 a_0$, gdzie a_i są cyfry. Wtedy

$$\begin{aligned} x &= a_0 + 10a_1 + \cdots + 10^n a_n = \\ &= T_0 + 1\,000T_1 + 1\,000\,000T_2 + 10^9 T_3 + \cdots + 10^{3k} T_k \\ &= T_0 + 1\,001T_1 - T_1 + 999\,999T_2 + T_2 + 1\,000\,000\,001T_3 - T_3 + \cdots \equiv \\ &\equiv T_0 - T_1 + T_2 - T_3 + \ldots \pmod{7(11, 13)}, \end{aligned}$$

bowiem $1001 = 7 \cdot 11 \cdot 13$ i $999\,999 = 3^3 \cdot 7 \cdot 11 \cdot 13 \cdot 37$.

Przykład 3.15. Liczba 47 528 481

1. jest podzielna przez 3, ponieważ $4+7+5+2+8+4+8+1 = 39 = 3 \cdot 13$,
2. nie jest podzielna przez 9.
3. jest podzielna przez 11, ponieważ $4-7+5-2+8-4+8-1 = 11$,
4. jest podzielna przez 7 i 13, ponieważ $47 - 528 + 481 = 0$.

3. Przykłady obliczeń w pierścieniu $\mathbb{Z}_m$

Przykład 3.16. W tym roku miałem urodziny we wtorek. W jakim dniu będę je miał za 10 lat?

Rozwiązanie. Modulo 7: $[2+10\cdot 365+2] = [2]+[10]\cdot[365]+[2] = [4]+[3]\cdot[1] = [0]$. Więc, w niedzielę.

Przykład 3.17. Zarabiam 17 złotych 28 groszy za godzinę. Ile będę miał drobnych za 2 143 godzin?

Rozwiązanie. Modulo 100: $[1\,728 \cdot 2\,143] = [28] \cdot [43] = \big([20]+[8]\big)\big([40]+[3]\big) = [60]+[20]+[24] = [4]$.

Przykład 3.18. Znaleźć ostatnią cyfrę liczby $2^{1\,000}$

Rozwiązanie. Ostatnia cyfra zgadza się z resztą przy dzieleniu przez 10. Więc modulo 10: $[2]^1 = [2]$, $[2]^2 = [4]$, $[2]^3 = [8]$, $[2]^4 = [6]$, $[2]^5 = [2]$. Ciąg potęg dwójki ma okres 4. W taki sposób,

$$[2^{1\,000}] = [2]^{1\,000} = [2]^{(1\,000 \mod 4)} = [2]^4 = 6.$$

4. Liczby pierwsze

Definicja 3.19. Liczba x, która ma dokładnie 2 podzielniki: 1 i x, nazywa się *pierwszą*. 1 nie jest liczbą pierwszą.

Twierdzenie 3.20 (Podstawowe twierdzenie arytmetyki). *Każda liczba naturalna $n > 1$ może być jednoznacznie rozłożona na czynniki*

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k},$$

gdzie $p_1 < p_2 < \cdots < p_k$ — liczby pierwsze, $\alpha_i > 0$ dla $i = 1, \ldots, k$.

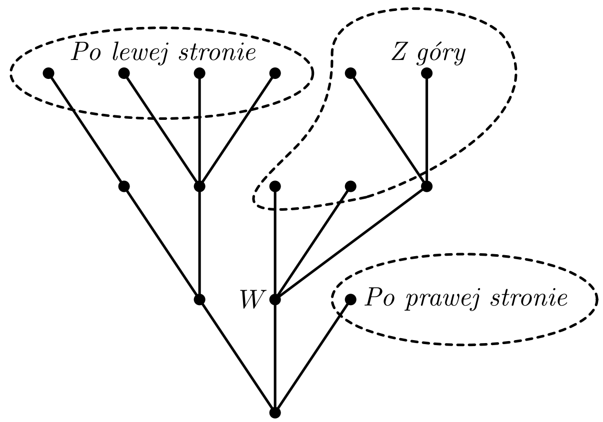
Algorytm poszukiwania liczb pierwszych. (Sito Eratostenesa).

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~	11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~	31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	~~49~~	~~50~~	~~51~~	~~52~~	53	~~54~~	~~55~~	~~56~~	~~57~~	~~58~~	59	~~60~~
61	~~62~~	~~63~~	~~64~~	~~65~~	~~66~~	67	~~68~~	~~69~~	~~70~~	71	~~72~~	73	~~74~~	~~75~~	~~76~~	~~77~~	~~78~~	79	~~80~~
~~81~~	~~82~~	83	~~84~~	~~85~~	~~86~~	~~87~~	~~88~~	89	~~90~~	~~91~~	~~92~~	~~93~~	~~94~~	~~95~~	~~96~~	97	~~98~~	~~99~~	~~100~~

W algorytmie 3.1 przedstawiony jest sposób obliczania S pierwszych liczb pierwszych poprzez dzielenie przez liczby pierwsze.

Rozwiązanie. Załóżmy, że obejście drzewa wykonuje Robot, znajdujący się na początku w korzeniu. Użyjmy następujących poleceń:

1. `WGórę` — przejście po pierwszej z wychodzących krawędzi (rysunek 5.2).
2. `WPrawo` — przejście do sąsiedniego po prawej stronie wierzchołka (rysunek 5.3).
3. `WDół` — zejście o jeden poziom w dól (rysunek 5.4).
4. `JestZGóru` — ma wartość **prawda**, jeśli istnieje wierzchołek z góry i **fałsz**, jeśli z góry nie ma wierzchołka.
5. `JestZPrawej` — ma wartość **prawda**, jeśli istnieje wierzchołek z prawej strony i **fałsz**, jeśli z prawej strony nie ma wierzchołka.
6. `JestZDołu` — ma wartość **prawda**, jeśli istnieje wierzchołek z dołu i **fałsz**, jeśli z dołu nie ma wierzchołka.



Rysunek 5.5. Klasy liści

Niech Robot będzie w pewnym wierzchołku W. Podzielmy wszystkie liście na klasy (rysunek 5.5):

1. *Po lewej stronie* — ścieżka z korzenia do liścia skręca w lewo przed wierzchołkiem W.
2. *Po prawej stronie* — ścieżka z korzenia do liścia skręca w prawo przed wierzchołkiem W.
3. *Z góry* — ścieżka z korzenia do liścia idzie przez wierzchołek W.

W szczególności, jeżeli W jest liściem, to on należy do klasy *„z góry”*. Dla korzenia klasy *„po lewej stronie”* oraz *„po prawej stronie”* są puste. Natomiast klasa *„z góry”* zgadza się ze zbiorem wszystkich liści.

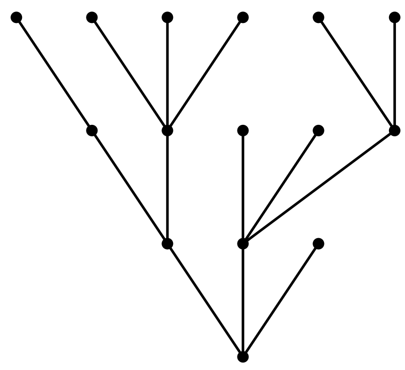
Zostanie potrzebna procedura `WGóręIWydrukować`, algorytm 5.1.

Program główny dany jest w algorytmie 5.2

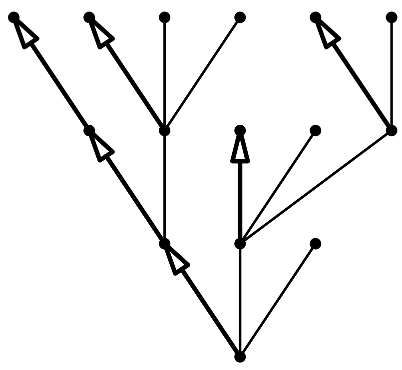
V. Wybrane algorytmy na grafach

1. Obejście drzewa poszukiwań (szukanie z powracaniem, backtraking)

Definicje 5.1. *Drzewem poszukiwań,* o n wierzchołkach nazywa się graf pusty, jeżeli $n = 0$, oraz, przy $n > 1$, graf, mający wierzchołek W, nazywany *korzeniem*, oraz skończoną (być może pustą) uporządkowaną ilość *poddrzew* poszukiwań $t_1, \ldots, t_k$ o $n_1, \ldots, n_k$ wierzchołkach odpowiednio. Przy czym $n_1 + \ldots + n_k + 1 = n$, (v. rysunek 5.1). Wierzchołek, który nie ma poddrzew, nazywa się *liściem*.

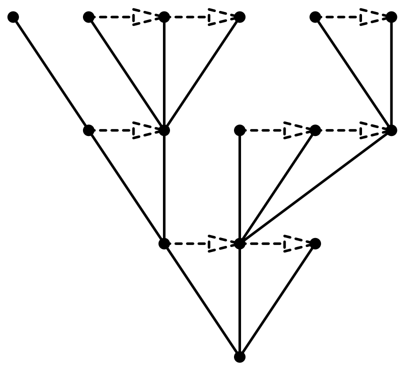


Rysunek 5.1. Drzewo poszukiwań

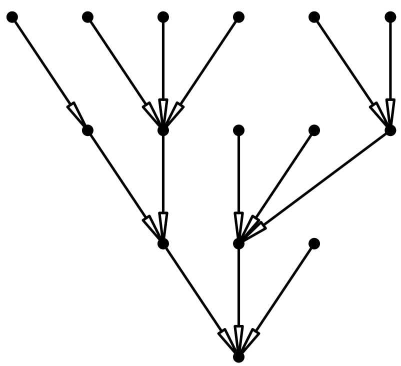


Rysunek 5.2. Polecenie `WGórę`

Zadanie 5.2. Dane jest drzewo poszukiwań. Wydrukować wszystkie jego liście dokładnie jeden raz.



Rysunek 5.3. Polecenie `WPrawo`



Rysunek 5.4. Polecenie `WDół`

Dane: $S > 2$
Wyniki: Tablica $P[0 \ldots S-1]$ zawiera S pierwszych liczb pierwszych
  $P[0] \leftarrow 2$
  $P[1] \leftarrow 3$
  $C \leftarrow 5$ {*Liczba C jest kandydatem na kolejną liczbę pierwszą*}
  $k \leftarrow 2$
  {*Niezmiennik: w tablicy P już zapisano k pierwszych liczb pierwszych*}
  while $k < S$ **do**
    $l \leftarrow 1$
    CisPrime $\leftarrow$ **true**
    {*Niezmiennik:* (*CisPrime* = **true** *i C nie jest podzielną przez* $P[0], \ldots,$ $P[l-1]$) *lub CisPrime* = **false**.}
    while *CisPrime* i $P[l] \leqslant \sqrt{C}$ **do**
      if C jest podzielna przez $P[l]$ **then**
        CisPrime $\leftarrow$ **false**
      else {*C nie jest podzielna przez* $P[l]$}
        $l \leftarrow l + 1$
      end if
    end while
    if *CisPrime* **then**
      $P[k] \leftarrow C$
      $k \leftarrow k + 1$
    end if
    $C \leftarrow C + 2$ {*Następny kandydat na liczbę pierwszą*}
  end while

Algorytm 3.1. Obliczenie pierwszych S liczb pierwszych

5. Największy wspólny podzielnik i najmniejsza wspólna wielokrotność

Przez $\mathrm{NWP}(x, y)$ oznaczamy największy wspólny podzielnik liczb x i y, przez $\mathrm{NWW}(x, y)$ — najmniejszą wspólną wielokrotność.

Definicja 3.21. Liczby x i y są *wzajemnie pierwsze,* jeśli $\mathrm{NWP}(x, y) = 1$.

Jako wniosek z twierdzenia 3.20 otrzymamy lemat:

Lemat 3.22. *Niech* $a = p_1^{\alpha_1} \ldots p_n^{\alpha_n}$ *oraz* $b = p_1^{\beta_1} \ldots p_n^{\beta_n}$ *będą rozłożenia liczb a i b na czynniki pierwsze (pozwalamy zerowe potęgi). Wtedy*

$$\mathrm{NWP}(a, b) = p_1^{\gamma_1} \ldots p_n^{\gamma_n}, \quad \mathrm{NWW}(a, b) = p_1^{\delta_1} \ldots p_n^{\delta_n},$$
$$\mathrm{NWW}(a, b) \cdot \mathrm{NWP}(a, b) = a \cdot b,$$

gdzie $\gamma_i = \min\{\, \alpha_i, \beta_i \,\}$, $\delta_i = \max\{\, \alpha_i, \beta_i \,\}$, $i = 1, \ldots n$.

Znany algorytm obliczania największego wspólnego podzielnika jest oparty na następującym twierdzeniu (algorytm 3.2).

Twierdzenie 3.23 (Euklides). *Niech $x > y$. Wtedy*

$$\mathrm{NWP}(x, y) = \mathrm{NWP}(x - y, y) = \mathrm{NWP}(x \bmod y, y).$$

Przykład 3.24.

$$\begin{aligned}\mathrm{NWP}(28, 49) = \mathrm{NWP}(28, 21) = \mathrm{NWP}(7, 21) = \\ = \mathrm{NWP}(7, 14) = \mathrm{NWP}(7, 7) = 7.\end{aligned}$$

Przykład 3.25. $\mathrm{NWP}(28, 124) = \mathrm{NWP}(28, 4) = \mathrm{NWP}(0, 4) = 4.$

Dane: $a > 0$, $b > 0$
Wyniki: $d = \mathrm{NWP}(a, b)$
  $m \leftarrow a$
  $n \leftarrow b$
  $\{$*Niezmiennik:* $NWP(a, b) = NWP(m, n),\ m \geqslant 0,\ n \geqslant 0\}$
  while $m \neq 0$ lub $n \neq 0$ **do**
    if $m > n$ **then**
      $m \leftarrow m \mod n$
    else $\{m \leqslant n\}$
      $n \leftarrow n \mod m$
    end if
  end while
  if $m = 0$ **then**
    $d \leftarrow n$
  else $\{n = 0\}$
    $d \leftarrow m$
  end if

Algorytm 3.2. Algorytm Euklidesa

Modyfikacja algorytmu 3.2 pozwala jednocześnie znaleźć takie liczby całkowite x i y, że $ax + by = \mathrm{NWP}(a, b)$, patrz algorytm 3.3.

Jeszcze jedna modyfikacja, zaproponowana przez Djikstrę, pozwala obliczyć jednocześnie $\mathrm{NWP}(a, b)$ i $\mathrm{NWW}(a, b)$, p. algorytm 3.4.

73. Sprawdzić, czy graf, określony macierzą przyległości $\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ ma ścieżkę Hamiltona?
74. Sprawdzić, czy graf skierowany, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$ ma cykle?
75. Sprawdzić, czy graf, określony macierzą incydencji $\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$ ma ścieżkę Hamiltona?
76. Sprawdzić, czy graf, określony macierzą incydencji $\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$ ma ścieżkę Eulera?
77. Sprawdzić, czy graf, określony macierzą incydencji $\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ jest eulerowskim?
78. Sprawdzić, czy graf, określony macierzą incydencji $\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ jest hamiltonowskim?
79. Sprawdzić, czy graf skierowany, określony macierzą incydencji $\begin{pmatrix} 1 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$ ma cykle?

61. Podać przykład grafa nie zorientowanego i odpowiedniej macierzy przyległości.
62. Podać przykład grafa nie zorientowanego i odpowiedniej macierzy incydencji.
63. Udowodnij, że suma elementów wiersza i macierzy przyległości jest równa $\delta(a_i)$.
64. Udowodnij, że suma elementów kolumny i macierzy przyległości jest równa $\delta(a_i)$.
65. Udowodnij, że suma elementów kolumny i macierzy przyległości grafa skierowanego jest równa $\delta^-(a_i)$.
66. Udowodnij, że suma elementów wiersza i macierzy przyległości grafa zorientowanego jest równa $\delta^+(a_i)$.
67. Udowodnij, że suma wierszy macierzy incydencji grafa zorientowanego jest równa wierszu zerowemu.
68. Sprawdzić, czy będzie graf, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$ eulerowskim?
69. Sprawdzić, czy będzie graf, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ spójnym?
70. Sprawdzić, czy graf, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ ma cykle?
71. Sprawdzić, czy będzie graf, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ hamiltonowskim?
72. Sprawdzić, czy graf, określony macierzą przyległości $\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$ ma ścieżkę Eulera?

Dane: $a > 0$, $b > 0$
Wyniki: $d = \mathrm{NWP}(a, b)$, $ax + by = d$

```
m ← a
n ← b
p ← 1
t ← 0
r ← 0
s ← 1
{Niezmiennik: NWP(a,b) = NWP(m,n), m ⩾ 0, n ⩾ 0, m = ap + bt,
  n = ar + bs}
while n ≠ 0 lub m ≠ 0 do
  if m > n then
    q ← m div n
    m ← m mod n
    p ← p − qr
    t ← t − qs
  else {m ⩽ n}
    q ← n div m
    n ← n mod m
    r ← r − qp
    s ← s − qt
  end if
end while
if m = 0 then
  d ← n
  x ← r
  y ← s
else {n = 0}
  d ← m
  x ← p
  y ← t
end if
```

Algorytm 3.3. Algorytm Euklidesa z poszukiwaniem liczb x, y, takich że $ax + by = \mathrm{NWP}(a, b)$

6. Pytania na egzamin

1. Niech będzie $a, b \in \mathbb{Z}$. Co nazywa się resztą przy dzieleniu a przez b?
2. Niech będzie $a, b \in \mathbb{Z}$. Co nazywa się ilorazem przy dzieleniu a przez b?
3. Sformułować twierdzenie o możliwości dzielenia liczb całkowitych.
4. Podzielić -17 przez -3 z resztą.
5. Co to znaczy, że x jest podzielnikiem y?
6. Co to znaczy, że x jest podzielna przez y?

Dane: $a > 0$, $b > 0$
Wyniki: $d = \mathrm{NWP}(a, b)$, $z = 2\,\mathrm{NWW}(a, b)$

$m \leftarrow a$
$n \leftarrow b$
$u \leftarrow b$
$v \leftarrow a$
$\{$*Niezmiennik:* $\mathrm{NWP}(a, b) = \mathrm{NWP}(m, n)$, $m \geqslant 0$, $n \geqslant 0$, $mu + nv = \mathrm{const}\}$
while $n \neq 0$ lub $m \neq 0$ **do**
  if $m > n$ **then**
    $v \leftarrow v + m \operatorname{div} n$
    $m \leftarrow m \mod n$
  else $\{m \leqslant n\}$
    $u \leftarrow u + n \operatorname{div} m$
    $n \leftarrow n \mod m$
  end if
end while
if $m = 0$ **then**
  $d \leftarrow n$
  $z \leftarrow v$
else $\{n = 0\}$
  $d \leftarrow m$
  $z \leftarrow u$
end if

Algorytm 3.4. Poszukiwanie jednocześnie $\mathrm{NWP}(a, b)$ oraz $\mathrm{NWW}(a, b)$

7. Podaj przykład dwóch liczb całkowitych x i y, takich że $x \mid y$.
8. Jakie dwie liczby nazywają się porównywalnymi modulo m?
9. Podaj przykład dwóch liczb całkowitych x i y, takich że $x \equiv y \pmod{17}$.
10. Podaj przykład liczby całkowitej x, takiej że $1 \equiv 18 \pmod{x}$.
11. Podaj przykład liczby całkowitej x, takiej że $1 \equiv -18 \pmod{x}$.
12. Niech liczby x i y będą miały jednakowe reszty przy dzieleniu przez m ($m > 1$). Udowodnij, że $x \equiv y \pmod{m}$.
13. Niech liczby x i y będą porównywalne modulo m ($m > 1$). Udowodnij, że x i y mają jednakowe reszty przy dzieleniu przez m.
14. Czy relacja porównywalności jest relacją równoważności?
15. Udowodnij, że relacja porównywalności jest zwrotną.
16. Udowodnij, że relacja porównywalności jest symetryczną.
17. Udowodnij, że relacja porównywalności jest przechodnią.
18. Co to jest „klasa reszt modulo m"?
19. Rozważmy klasy reszt modulo 5. Czy $[-3] = [2]$? Dlaczego?
20. Niech będzie $x_1 \equiv x_2 \pmod{m}$ oraz $y_1 \equiv y_2 \pmod{m}$. Udowodnij, że $x_1 + y_1 \equiv x_2 + y_2 \pmod{m}$.

22. Niech $G = (A, B)$ będzie grafem zorientowanym. Udowodnić, że $\sum_{a \in A} \delta^-(a) = |B|$.
23. Niech $G = (A, B)$ będzie grafem zorientowanym. Udowodnić, że $\sum_{a \in A} \delta^-(a) = \sum_{a \in A} \delta^+(a)$.
24. Czy graf może mieć dokładnie 5 wierzchołków o stopniach 1, 2, 3, 4 oraz 5?
25. Ile krawędzi ma zupełny graf o n węzłach?
26. Co nazywa się marszrutą w grafie?
27. Podać przykład marszruty w grafie.
28. Podać przykład cyklu w grafie.
29. Co nazywa się cyklem w grafie?
30. Co nazywa się grafem acyklicznym?
31. Jaki graf nazywa się drzewem?
32. Co nazywa się długością grogi w grafie?
33. Jaki graf nazywa się spójnym?
34. Podać przykład spójnego grafa.
35. Podać przykład grafa nie spójnego.
36. Co jest składową grafa?
37. Podaj przykład grafa o 3 składowych.
38. Udowodnić, że jeśli graf ma dokładnie dwa wierzchołki o nieparzystym stopniu, to istnieje ścieżka, łącząca te dwa wierzchołki.
39. Jaka ścieżka w grafie nazywa się ścieżką Eulera?
40. Podać definicję cyklu Eulera.
41. Jaki graf nazywa się eulerowskim?
42. Podać przykład eulerowskiego grafa.
43. Podać przykład grafa nie eulerowskiego.
44. Jaki jest konieczny i dostateczny warunek istnienia w grafie cyklu Eulera?
45. Udowodnić, że każdy wierzchołek eulerowskiego grafa ma parzystą stopień.
46. Jaki jest konieczny i dostateczny warunek istnienia w grafie ścieżki Eulera?
47. Niech graf Γ będzie miał ścieżkę Eulera, łączącą węzły a i b ($a \neq b$). Udowodnij, że węzły te są jedyne węzły o nieparzystym stopniu.
48. Jaka ścieżka w grafie nazywa się ścieżką Hamiltona?
49. Podać definicję cyklu Hamiltona.
50. Jaki graf nazywa się hamiltonowskim?
51. Podać przykład hamiltonowskiego grafa.
52. Podać przykład grafa nie hamiltonowskiego.
53. Podać dostateczny warunek istnienia w grafie cyklu Hamiltona.
54. Udowodnij, że graf zupełny jest hamiltonowskim.
55. Podać definicję macierzy incydencji grafa zorientowanego.
56. Podać definicję macierzy incydencji grafa nie zorientowanego.
57. Podać definicję macierzy przyległości grafa zorientowanego.
58. Podać definicję macierzy przyległości grafa nie zorientowanego.
59. Podać przykład grafa zorientowanego i odpowiedniej macierzy przyległości.
60. Podać przykład grafa zorientowanego i odpowiedniej macierzy incydencji.

5. Hamiltonowskie ścieżki i cykle w grafach

Definicje 4.12.

1. *ścieżką hamiltonowską* w grafie nazywa się ścieżka, przechodząca przez każdy wierzchołek dokładnie jeden raz.
2. *Cyklem hamiltonowskim* w grafie nazywa się cykl, przechodzący przez każdy wierzchołek dokładnie jeden raz.
3. Graf nazywa się *hamiltonowskim*, jeżeli posiada on cykl hamiltonowski.

Twierdzenia 4.13.

1. *Graf zupełny jest hamiltonowskim.*
2. *Jeżeli dla każdej pary wierzchołków a i b grafa Γ o m wierzchołkach bez pętel $\delta(a) + \delta(b) \geqslant m$, to graf Γ jest hamiltonowskim.*
3. *Jeżeli dla każdego wierzchołka a grafa Γ o m wierzchołkach bez pętel zachodzi $\delta(a) \geqslant \frac{m}{2}$, to graf Γ jest hamiltonowskim.*

Dowody.

1. Indukcja po ilości wierzchołków.
3. Wniosek z twierdzenia 2.

6. Pytania na egzamin

1. Podaj definicję grafa.
2. Podaj definicję grafa zorientowanego.
3. Podaj przykład grafa o 6 wierzchołkach i 3 krzwędziach.
4. Podaj przykład grafa z pętlą.
5. Podaj przykład grafa, który ma krawędzie równoległe.
6. Podaj przykład grafa zorientowanego o 3 wierzchołkach i 6 łukach.
7. Jakie dwa wierzchołki nazywają się sąsiednimi?
8. Zdefiniuj pojęcie incydencji.
9. Co jest stopniem wierzchołka grafa?
10. Jaki węzeł grafa nazywa się izolowanym?
11. Jaki węzeł grafa nazywa się wiszącym?
12. Jaki łuk (krawędź) grafa nazywa się pętlą?
13. Jaki graf nazywa się zupełnym?
14. Podać przykład grafa zupełnego.
15. Co jest półstopniem wyjścia wierzchołka grafa zorientowanego?
16. Co jest półstopniem wejścia wierzchołka grafa zorientowanego?
17. Ile wynosi suma stopni wszystkich węzłów grafa?
18. Ile wynosi suma półstopni wyjścia wszystkich węzłów grafa?
19. Ile wynosi suma półstopni wejścia wszystkich węzłów grafa?
20. Niech $G = (A, B)$ będzie grafem. Udowodnić, że $\sum_{a\in A} \delta(a) = 2|B|$.
21. Niech $G = (A, B)$ będzie grafem zorientowanym. Udowodnić, że $\sum_{a\in A} \delta^+(a) = |B|$.

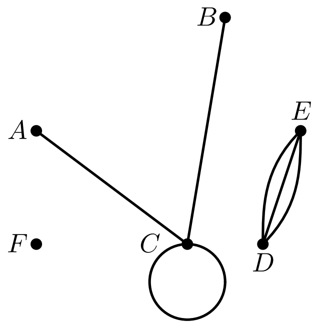
21. Niech będzie $x_1 \equiv x_2 \pmod{m}$ oraz $y_1 \equiv y_2 \pmod{m}$. Udowodnij, że $x_1 \times y_1 \equiv x_2 \cdot y_2 \pmod{m}$.
22. Niech będzie $x_1 \equiv x_2 \pmod{m}$ oraz $y_1 \equiv y_2 \pmod{m}$. Udowodnij, że $x_1 - y_1 \equiv x_2 - y_2 \pmod{m}$.
23. Podaj definicję sumy klas reszt.
24. Podaj definicję iloczynu klas reszt.
25. Dlaczego relacja porównywalności dzieli zbiór $\mathbb{Z}$ na nieprzecinające się klasy?
26. Rozważmy klasy reszt modulo 5. Czy $-3 \in [2] + [100]$? Dlaczego?
27. Rozważmy klasy reszt modulo 5. Znaleźć x, takie że $[x] \cdot [3] = [-1]$.
28. Udowodnij, że $[x] + [y] = [y] + [x]$.
29. Udowodnij, że $[x] + ([y] + [z]) = ([x] + [y]) + [z]$.
30. Udowodnij, że $[x] + [0] = [0] + [x] = [x]$.
31. Udowodnij, że $[x] + [-x] = [0]$.
32. Udowodnij, że $[x] \cdot [y] = [y] \cdot [x]$.
33. Udowodnij, że $[x] \cdot ([y] \cdot [z]) = ([x] \cdot [y]) \cdot [z]$.
34. Udowodnij, że $[x] \cdot [1] = [1] \cdot [x] = [x]$.
35. Udowodnij, że $[x] \cdot ([y] + [z]) = [x] \cdot [y] + [x] \cdot [z]$.
36. Udowodnij, że $([x] + [y]) \cdot [z] = [x] \cdot [z] + [y] \cdot [z]$.
37. Co nazywa się pierścieniem $\mathbb{Z}_m$?
38. Rozważmy klasy reszt modulo 5. Oblicz $[192\,873] \cdot [-12\,123\,130\,397] + [1\,212\,336] \cdot [374\,112\,341]$.
39. W roku 2003 Boże Narodzenie było w czwartek. W jaki dzień było Boże Narodzenie w roku 1903?
40. Znaleźć ostatnią cyfrę liczby $3^{2\,003}$.
41. Jakie jest kryterium podzielności przez 3?
42. Udowodnij kryterium podzielności przez 3.
43. Jakie jest kryterium podzielności przez 9?
44. Udowodnij kryterium podzielności przez 9.
45. Jakie jest kryterium podzielności przez 11?
46. Udowodnij kryterium podzielności przez 11.
47. Jakie jest kryterium podzielności przez 13?
48. Udowodnij kryterium podzielności przez 13.
49. Jakie jest kryterium podzielności przez 7?
50. Udowodnij kryterium podzielności przez 7.
51. Przez jakie z liczb 2, 3, 5, 6, 7, 9, 10, 11 oraz 13 jest podzielna liczba 192 827 364 548 495 066 253 444?
52. Jaka liczba nazywa się pierwszą?
53. Podaj pięć liczb pierwszych.
54. Czy liczba pierwsza może być podzielna przez 5?
55. Sformułuj podstawowe twierdzenie arytmetyki.
56. Na czym polega algorytm „Sito Eratostenesa”?
57. Co nazywa się największym wspólnym podzielnikiem dwóch liczb całkowitych?

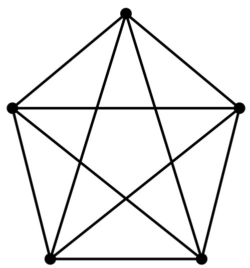
58. Co nazywa się najmniejszą wspólną wielokrotnością dwóch liczb całkowitych?
59. Jakie liczby nazywają się wzajemnie-pierwszymi?
60. Podaj przykład liczb wzajemnie-pierwszych?
61. Podaj przykład liczb, które nie są wzajemnie-pierwsze?
62. Niech $a = p_1^{\alpha_1} \dots p_n^{\alpha_n}$ oraz $b = p_1^{\beta_1} \dots p_n^{\beta_n}$ będą rozłożenia liczb a i b na czynniki pierwsze (pozwalamy zerowe potęgi). Jakim będzie rozłożenie $\mathrm{NWP}(a, b)$?
63. Niech $a = p_1^{\alpha_1} \dots p_n^{\alpha_n}$ oraz $b = p_1^{\beta_1} \dots p_n^{\beta_n}$ będą rozłożenia liczb a i b na czynniki pierwsze (pozwalamy zerowe potęgi). Jakim będzie rozłożenie $\mathrm{NWW}(a, b)$?
64. Sformułuj twierdzenie Euklidesa o największym wspólnym podzielniku.
65. Oblicz $\mathrm{NWP}(35, 42)$.
66. Oblicz $\mathrm{NWW}(35, 42)$.
67. Rozważmy algorytm Euklidesa obliczenia $\mathrm{NWP}(a, b)$ dla liczb 42 i 35. Podaj pierwsze trzy kroki tego algorytmu.
68. Rozważmy algorytm Euklidesa obliczenia $\mathrm{NWP}(a, b)$ i liczb x, y, takich, że $ax + by = \mathrm{NWP}(a, b)$. Jakie będą znalezione liczby x i y, jeżeli $a = 35$, $b = 42$?
69. Rozważmy modyfikację Djikstry algorytmu Euklidesa obliczenia $\mathrm{NWP}(a, b)$ i $\mathrm{NWW}(a, b)$ dla liczb 42 i 35. Podaj pierwsze trzy kroki tego algorytmu.

IV. Teoria grafów

1. Podstawowe definicje

Uwaga 4.1. Teoria grafów nie ma ustalonego języka — każdy autor używa swojej terminologii. Określone w tym rozdziale pojęcia mogą się różnić od analogicznych, używanych w innych książkach. Starałem się użyć takiej terminologii, która pozwala jednoznacznie traktować określone obiekty.





Rysunek 4.1. Przykład grafa   Rysunek 4.2. Graf zupełny

Twierdzenie 4.9 (Własności macierzy przyległości i incydencji). *Niech Γ będzie grafem bez pętel i krotnych krawędzi (łuków).*

1. *Dla grafa nieskierowanego Γ suma elementów macierzy $S(\Gamma)$ w wierszu i albo w kolumnie i równa jest $\delta(a_i)$.*
2. *Dla grafa skierowanego Γ suma elementów macierzy $S(\Gamma)$ w kolumnie i równa jest $\delta^+(a_i)$, zaś suma elementów macierzy w wierszu i jest równa $\delta^-(a_i)$.*
3. *Dla grafa nieskierowanego Γ suma wierszy macierzy $C(\Gamma)$ jest równa wierszu o samych dwójkach.*
4. *Dla grafa skierowanego Γ suma wierszy macierzy $C(\Gamma)$ jest równa wierszu zerowemu.*
5. Element s_{ij}^k macierzy $S^k(\Gamma)$ równy jest ilości ścieżek o długości k łączących a_i z a_j.
6. Na to, żeby graf skierowany Γ o n wierzchołkach bez pętel miał przynajmniej jeden cykl potrzeba i wystarczy, żeby macierz $S^2(\Gamma) + S^3(\Gamma) + \dots + S^n(\Gamma)$ miała niezerowe elementy na przekątnej.
7. Na to, żeby nieskierowany graf Γ o n wierzchołkach miał przynajmniej jeden cykl potrzeba i wystarczy, żeby macierz $S(\Gamma) + S^2(\Gamma) + S^3(\Gamma) + \dots + S^n(\Gamma)$ miała niezerowe elementy na przekątnej.

4. Grafy eulerowskie

Rozważmy w tym rozdziale grafy bez wierzchołków izolowanych.

Definicje 4.10.

1. *Ścieżką euleroswką* w grafie nazywa się ścieżka, zawierająca dokładnie jeden raz wszystkie krawędzie grafa.
2. *Eulerowskim cyklem* w Grafie nazywa się cykl, zawierający dokładnie jeden raz wszystkie krawędzie grafa.
3. Graf nazywa się *eulerowskim*,jeżeli ma on eulerowski cykl, v. rysunek 4.5.

Twierdzenia 4.11.

1. *Eulerowski graf jest spójny i wszystkie jego wierzchołki mają parzysty stopień.*
2. *Jeżeli graf Γ jest spójnym i wszystkie jego wierzchołki mają parzysty stopień, to graf ten jest eulerowski.*
3. *Jeżeli graf Γ ma ścieżkę eulerowską o końcach a i b, $(a \neq b)$, to graf ten jest spójnym oraz a i b są jedynymi wierzchołkami o nieparzystym stopniu.*
4. *Jeżeli graf Γ jest spójnym i ma dokładnie dwa wierzchołki a i b $(a \neq b)$ o nieparzystym stopniu, to graf Γ ma ścieżkę eulerowską o końcach a i b.*

Definicje 4.8.

1. *Macierzą przyległości* grafa Γ nazywa się macierz $S(\Gamma) = (s_{ij})$, $i = 1, \ldots, n$, $j = 1, \ldots, n$, gdzie
$$s_{ij} = \begin{cases} 1, & \text{jeżeli wierzchołki } a_i \text{ i } a_j \text{ połączone są krawędzią,} \\ 0, & \text{jeżeli nie istnieje krawędzi, łączącej wierzchołki } a_i \text{ i } a_j. \end{cases}$$
2. *Macierzą incydencji* grafa Γ nazywa się macierz $C(\Gamma) = (c_{ij})$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, gdzie
$$c_{ij} = \begin{cases} 1, & \text{jeżeli wierzchołek } a_i \text{ jest incydentny z krawędzią } j, \\ 0, & \text{jeżeli wierzchołek } a_i \text{ nie jest incydentny z krawędzią } j. \end{cases}$$
3. *Macierzą przyległości* grafa skierowanego Γ nazywa się macierz $S(\Gamma) = (s_{ij})$, $i, j = 1, \ldots, n$, gdzie
$$s_{ij} = \begin{cases} 1, & \text{jeżeli istnieje łuk } (a_i, a_j), \\ 0, & \text{jeżeli nie istnieje łuku } (a_i, a_j). \end{cases}$$
4. *Macierzą incydencji* grafa skierowanego Γ nazywa się macierz $C(\Gamma) = (c_{ij})$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, gdzie
$$c_{ij} = \begin{cases} 1, & \text{jeżeli wierzchołek } a_i \text{ jest końcem łuku } j, \\ -1, & \text{jeżeli wierzchołek } a_i \text{ jest początkiem łuku } j, \\ 0, & \text{jeżeli wierzchołek } a_i \text{ nie jest incydentny z łukiem } j. \end{cases}$$

Przykłady.

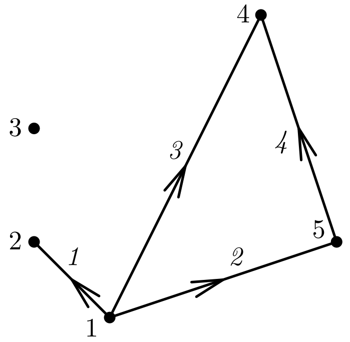
1. Dla grafa na rysunku 4.4 macierze przyległości oraz incydencji będą następujące:
$$S = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$
2. Dla grafa skierowanego na rysunku 4.3 odpowiednio:
$$S = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

Definicje 4.2.

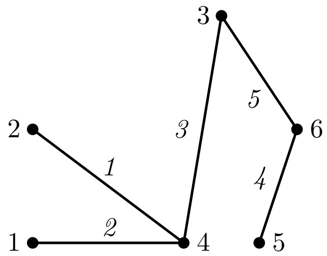
1. Zbiór $\mathfrak{M} = \{\, a_1, a_2, \ldots \}$ i zestaw[1] $\mathfrak{N}$ nieuporządkowanych par elementów $v_k = (a_{i_k} a_{j_k})$ $\mathfrak{M}$ nazywa się *grafem* $\Gamma = (\mathfrak{M}, \mathfrak{N})$. Elementy zbioru $\mathfrak{M}$ nazywają się *wierzchołkami (albo węzłami),* a obiekty z zestawu $\mathfrak{N}$ nazywają się *krawędziami* grafa Γ. Cf. rysunek 4.1, gdzie
$$\mathfrak{M} = \{\, A, B, D, D, E, F \,\}, \quad \mathfrak{N} = \{\, AC, CC, CB, DE, DE, DE \,\}$$
2. Mówią, że krawędź $v = (a_i a_j)$ *łączy* wierzchołki a_i i a_j.
3. Mówimy, że wierzchołki a_i i a_j są *końcami* krawędzi $v = (a_i a_j)$.
4. Jeżeli wierzchołek a należy do krawędzi v, to mówimy, że one są *incydentne*.
5. Ilość krawędzi, incydentnych wierzchołku a nazywa się jego *stopniem*, oznacza się $\delta(a)$. Na przykład, na rysunku 4.1 $\delta(A) = 1$, $\delta(B) = 1$, $\delta(C) = 4$, $\delta(D) = 3$, $\delta(E) = 3$, $\delta(F) = 0$.
6. Wierzchołek o stopniu 0 nazywa się *izolowanym*, wierzchołek o stopniu 1, — *wiszącym.* Cf. rysunek 4.1, gdzie wierzchołek F jest izolowany, zaś wierzchołki A i B są wiszące.
7. Dwie krawędzie są *przyległe,* jeżeli mają one wspólny wierzchołek.
8. Dwa wierzchołki są *sąsiednie,* jeżeli są one połączone krawędzią.
9. Krawędź postaci (a_i, a_i) nazywa się *pętlą.* Cf. rysunek 4.1, gdzie CC jest pętlą.
10. Krawędzie, zawierające się w $\mathfrak{N}$ kilka razy, są *wielokrotne (równoległe).* Ilość wchodzeń krawędzi w $\mathfrak{N}$ nazywa się *krotnością.* Cf. rysunek 4.1, gdzie krotność krawędzi ED jest równa 3, zaś pozostałe krawędzie mają krotność 1.
11. Graf bez krotnych krawędzi nazywa się *prostym.*
12. Jeżeli wierzchołki, definiujące krawędź grafa Γ, są uporządkowane, to graf nazywa się *skierowanym (zorientowanym)*, a jego krawędzie nazywają się *łukami.* Cf. rysunek 4.3.
13. O łuku $v = (a_i a_j)$ mówią, że *wychodzi* on z wierzcholka a_i i *wchodzi* w wierzchołek a_j.
14. Wierzchołek a_i nazywa się *początkiem,* zaś wierzchołek a_j — *końcem* łuku $v = (a_i a_j)$.
15. Ilość łuków, wychodzących z wierzchołka a, nazywa się *półstopniem wyjścia,* oznacza się przez $\delta^-(a)$. Na przykład, dla grafa z rysunku 4.3 $\delta^-(a_1) = 3$, $\delta^-(a_2) = 0$, $\delta^-(a_3) = 0$, $\delta^-(a_4) = 0$, $\delta^-(a_5) = 1$.
16. Ilość łuków, wchodzących do wierzchołka a, nazywa się *półstopniem wejścia,* oznacza się przez $\delta^+(a)$. Na przykład, dla grafa z rysunku 4.3 $\delta^+(a_1) = 0$, $\delta^+(a_2) = 1$, $\delta^+(a_3) = 0$, $\delta^+(a_4) = 2$, $\delta^+(a_5) = 1$.

[1] ten sam element może się zawierać w zestawie kilka razy

17. Graf jest *zupełnym,* jeżeli dwa jego dowolne wierzchołki połączone są krawędzią. V. rysunek 4.2.



Rysunek 4.3. Graf skierowany



Rysunek 4.4. Drzewo

Uwaga 4.3. Często grafy przedstawia się za pomocą rysunków, na których wierzchołkom odpowiadają punkty, zaś krawędziom (łukom) — łączące je krzywe. Przy tym krawędzie nie mają wspólnych punktów oprócz wierzchołków. V. rysunki 4.1–4.3

Twierdzenia 4.4. *Dla grafa* $\Gamma = \{\, \mathfrak{M}, \mathfrak{N} \,\}$

1. $\sum\limits_{a \in \mathfrak{M}} \delta(a) = 2|\mathfrak{N}|$.
2. *Jeżeli graf jest skierowany, to* $\sum\limits_{a \in \mathfrak{M}} \delta^{+}(a) = \sum\limits_{a \in \mathfrak{M}} \delta^{-}(a) = |\mathfrak{N}|$.

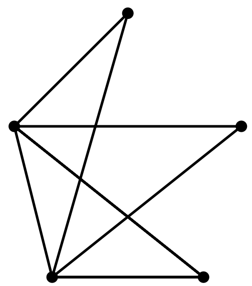
2. Drogi i cykle w grafach

Definicje 4.5.

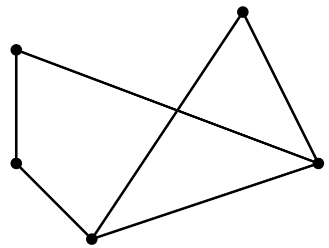
1. Układ krawędzi grafa Γ $A_{a_1 a_n} = \{\, (a_1, a_2), (a_2, a_3), \ldots, (a_{n-1}, a_n) \,\}$, nazywa się *marszrutą (trasą, drogą, ścieżką)*[2], łączącą wierzchołki a_1 i a_n. Zapisujemy marszrutę również $a_1 \to a_2 \to a_3 \to \cdots \to a_{n-1} \to a_n$.
2. Liczba krawędzi w marszrucie nazywa się jej *długością.*
3. ścieżka A_{a_i, a_j}, wszystkie krawędzie której są różne, nazywa się *cyklem*, jeśli $a_i = a_j$. Na przykład, graf na rysunku 4.1 ma cykle $C \to C$ oraz $D \to E \to D$.
4. Cykl w grafie nazywa się *prostym,* jeżeli wszystkie jego wierzchołki są różne.

[2] w niektórych książkach ścieżki określa się, jako marszruty, wszystkie krawędzie której są różne, zaś drogi — jako ścieżki o różnych wierzchołkach, cf. uwagę 4.1

5. Graf nazywa się *spójnym,* jeżeli dwa dowolne wierzchołki są połączone ścieżką.
6. Graf nazywa się *acyklicznym*, jeżeli on nie zawiera cykli.
7. Acykliczny spójny graf nazywa się *drzewem*, v. rysunek 4.4.
8. Graf $\Gamma_1 = \{\, \mathfrak{M}_1, \mathfrak{N}_1 \,\}$ jest *podgrafem* grafa $\Gamma = \{\, \mathfrak{M}, \mathfrak{N} \,\}$ jeżeli $\mathfrak{M}_1 \subset \mathfrak{M}$, oraz $\mathfrak{N}_1 \subset \mathfrak{N}$.
9. Spójny podgraf $\Gamma_1 = \{\, \mathfrak{M}_1, \mathfrak{N}_1 \,\}$ nazywa się *składową* grafa $\Gamma = \{\, \mathfrak{M}, \mathfrak{N} \,\}$, jeżeli żaden z wierzchołków z $\mathfrak{M} \setminus \mathfrak{M}_1$ nie jest połączony krawędzią z żadnym z wierzchołków z $\mathfrak{N}$. Na przykład, graf na rysunku 4.1 ma 3 składowe, graf na rysunku 4.2 — jedną, graf na rysunku 4.3 — dwie.



Rysunek 4.5. Graf eulerowski



Rysunek 4.6. Graf hamiltonowski

Twierdzenie 4.6. *Niech graf ma dokładnie dwa wierzchołki o nieparzystym stopniu. Wtedy istnieje ścieżka, łącząca te wierzchołki.*

Dowód. Zastosować twierdzenie 4.4, punkt 1 do składowych grafa.

Dalej w tym rozdziale każdy graf jest prostym bez pętel

3. Macierzowa reprezentacja grafa

Definicja 4.7. *Macierzą* $n \times m$ nazywamy prostokątną tablicę liczb:

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}.$$

Mówimy również, że macierz A ma n *wierszy* i m *kolumn.* Zapis a_{ik} oznacza, że element ten znajduje się w i-tym wierszu i k-tej kolumnie. Jeżeli $m = n$, macierz nazywa się *kwadratową stopnia* n.

Niech Γ będzie grafem o n wierzchołkach i m krawędziach.