

Algorytmy i struktury danych (studia stacjonarne)

Dr Anna Muranova

Semestr zimowy 2025/2026, UWM w Olsztynie

Ćwiczenie 4 dla **gr. 1** (Jeszcze raz rekurencja oraz sortowanie) 29.10.2025

Zadanie 4.1. Zaimplementować funkcję `n_sum(n: int, m: int) -> list[int]`, która zwróci wszystkie n -cyfrowe liczby z sumą cyfr $m > 0$ metodą nierekurencyjną.

Zadanie 4.2. Zaimplementować funkcję `n_sums(m: int, p: int, np: int) -> list[int]`, która zwróci wszystkie n -cyfrowe liczby z sumą cyfr p na indeksach parzystych i np na nieparzystych metodą rekurencyjną..

Zadanie 4.3. Zaimplementować funkcję `power(number: int, n: int) -> int`, której zadaniem jest zwrócenie wyniku działania $number^n$

- wprost,
- przez algorytm szybkiego potęgowania

https://pl.wikipedia.org/wiki/Algorytm_szybkiego_pot%C4%99gowania

NIE UŻYWAĆ OPERATORA `**`. Zaimplementować oba algorytmu rekurencyjnie i nierekurencyjne.

Zadanie 4.4. Zaimplementować funkcję `calculator(t: tuple) -> int`, która oblicze wyraz t podany w postaci trójelementowej krotki, w której skrajne elementy są wyrazami lub liczbami, a środkowe znakiem `"+"` lub `"*"`. rekurencyjnie i nierekurencyjnie. Przykład

`calculator ((5, '+', -1), '+', 3), '-', 2) -> 5`

Zadanie 4.5. Posortować jeden z poniższych zestawów liczb na kartce na każde z trzech sposobów:

- bąbelkowe (bubble sort), złożoność: $O(n^2)$
- przez wybieranie, złożoność: $O(n^2)$
- przez wstawianie (insertion sort) z zaawansowanym przesuwaniem, złożoność: $O(n^2)$

Zestawy liczb:

- 3, 4, 0, 7, 1, 5, 2, 8, 9, 6
- 3, 5, 8, 7, 1, 6, 9, 4, 0, 2
- 0, 7, 9, 4, 8, 6, 5, 3, 2, 1
- 4, 5, 2, 7, 0, 9, 1, 3, 6, 8
- 0, 8, 2, 7, 6, 3, 9, 1, 5, 4
- 2, 0, 7, 3, 4, 5, 8, 6, 9, 1