Jakbym rozwiązał zadanie 10830, gdybym był studentem

Alexander Denisjuk

3 kwietnia 2005 roku

Gdybym była. . .
Kayah

1 64-bitowe liczby

W zadaniu jest napisano:

> You can safely assume that any output number fits in a 64-bit signed integer.

Więc przede wszystkim, ja bym się zastanowił, dlaczego w Microsoft® Visual C++® 6.0 nie działa `long long int`. Po chwili zastanowienia ja bym odinstalował Microsoft® Visual C++® 6.0 oraz Microsoft® Windows®i skorzystałbym ze standardowego kompilatora C++. Na przykład, Gnu C++.

2 Obliczenie z definicji

Jest sprawą beznadziejną. Żeby się w tym przekonać, wystarczy napisać pętlę

```
n=2000000000;
S=0;
for (i=2;i<=n/2;i++){
    if (n % i ==0) S+=i;
}
```

i pomierzyć czas działania tej pętli. A takich pętel może być nawet 50!

3 Analiza

Więc trzeba zadanie to przeanalizować. Stworzymy tablicę

n	1	2	3	4	5	6	7	8	9	10	11	12	13
SOD(n)	0	0	0	2	0	5	0	6	3	7	0	15	0
CSOD(n)	0	0	0	2	2	7	7	13	16	23	23	38	38

I wpatrujemy się w nią przez dwie minuty.

4 Jak można obliczyć CSOD(n)?

Z czego składa się CSOD(13)? Na pewno każda parzysta liczba (oprócz dwójki) doda swój podzielnik-dwójkę. Każda liczba podzielna przez 3 (poza trójką) doda trójkę. I tak dalej.

Ile jest liczb parzystych, mniejszych od 13? Bez wątpienia $\left[\frac{13}{2}\right]$. Ile jest wielokrotności trójki, mniejszych od 13? Bez wątpienia $\left[\frac{13}{3}\right]$. I tak dalej.

A oto poszukiwany wzór (czemu suma tylko do $\left[\frac{n}{2}\right]$?):

$$\mathrm{CSOD}(n) = \sum_{i=2}^{\left[\frac{n}{2}\right]} \left(\left[\frac{n}{i}\right] - 1\right) \cdot i = \sum_{i=2}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i - \sum_{i=2}^{\left[\frac{n}{2}\right]} i =$$

$$= -\frac{(\left[\frac{n}{2}\right] + 2)(\left[\frac{n}{2}\right] - 1)}{2} + \sum_{i=2}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i. \quad (1)$$

We wzorze użyto oznaczenia $[x]$ — część całkowita x. Czyli największa liczba całkowita nie większa od x. Niestety, w najgorszym przypadku ($n = 2\,000\,000\,000$) suma (1) zawiera miliard składników i w taki sposób też się nie nadaje do obliczeń.

5 Jak można inaczej obliczyć sumę (1)?

Skoro suma jest za duża, by być obliczoną na komputerze, sprobujmy obliczyć ją częściowo ręcznie. Zauważmy, że przy zmianie i wyrażenie $\left[\frac{n}{i}\right]$ może się nie zmieniać. W rzeczywistości jeśli $\left[\frac{n}{l}\right] + 1 \leqslant i \leqslant \left[\frac{n}{l-1}\right]$, to $\left[\frac{n}{i}\right] = l - 1$ (czemu?).

Można rozbić sumę w (1) na części:

$$\sum_{i=2}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i = \sum_{l=\left[\frac{n}{2}\right]+1}^{3} \sum_{i=\left[\frac{n}{l}\right]+1}^{\left[\frac{n}{l-1}\right]} \left[\frac{n}{i}\right] \cdot i = \sum_{l=\left[\frac{n}{2}\right]+1}^{3} (l-1) \sum_{i=\left[\frac{n}{l}\right]+1}^{\left[\frac{n}{l-1}\right]} i =$$

$$= \sum_{l=\left[\frac{n}{2}\right]+1}^{3} \frac{(l-1)}{2} \left(\left[\frac{n}{l-1}\right] + \left[\frac{n}{l}\right] + 1\right) \left(\left[\frac{n}{l-1}\right] - \left[\frac{n}{l}\right]\right). \quad (2)$$

Niestety, otrzymany wzór zawiera tyle samo, co (1) składników i to bardziej skomplikowanych!

6 Kombinacja wzorów (1) i (2)

Osobliwością wzoru (2) jest to, że każdy składnik zastępuję sumę wzoru (1) na odcinku od $\left[\frac{n}{l}\right] + 1$ do $\left[\frac{n}{l-1}\right]$. Odcinek ten dla małych l (czyli na końcu sumy wzoru (1)) jest duży, natomiast dla dużych l (na początku sumy wzoru (1)) jest mały. Sprobujmy więc początek sumy wzoru (1) obliczyć bezpośrednio, zaś koniec zastąpić przez wzór (2):

$$\sum_{i=2}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i = \sum_{i=2}^{\left[\frac{n}{k}\right]} \left[\frac{n}{i}\right] \cdot i + \sum_{i=\left[\frac{n}{k}\right]+1}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i, \tag{3}$$

gdzie k jest pewna stałą, o której pomyślimy później.

Uprośćmy pierwszą sumę w (3):

$$\sum_{i=2}^{\left[\frac{n}{k}\right]} \left[\frac{n}{i}\right] \cdot i = \sum_{i=2}^{\left[\frac{n}{k}\right]} \left(\frac{n}{i} - \left\{\frac{n}{i}\right\}\right) \cdot i = \sum_{i=2}^{\left[\frac{n}{k}\right]} n - \sum_{i=2}^{\left[\frac{n}{k}\right]} n \,\%\, i = n\left(\left[\frac{n}{k}\right] - 1\right) - \sum_{i=2}^{\left[\frac{n}{k}\right]} n \,\%\, i,$$

gdzie $\{x\}$ jest częścią ułamkową liczby x, czyli $\{x\} = x - [x]$. Jako $m \,\%\, n$ oznaczono resztę przy dzieleniu m przez n. A propos, czemu $\left\{\frac{n}{i}\right\} = \frac{n \,\%\, i}{i}$?

Druga sumą zapisze się w sposób następujący:

$$\sum_{i=\left[\frac{n}{k}\right]+1}^{\left[\frac{n}{2}\right]} \left[\frac{n}{i}\right] \cdot i = \sum_{l=k}^{3} \sum_{i=\left[\frac{n}{l}\right]+1}^{\left[\frac{n}{l-1}\right]} \left[\frac{n}{i}\right] \cdot i ==$$

$$= \sum_{l=k}^{3} \frac{(l-1)}{2} \left(\left[\frac{n}{l-1}\right] + \left[\frac{n}{l}\right] + 1\right)\left(\left[\frac{n}{l-1}\right] - \left[\frac{n}{l}\right]\right).$$

Podsumując, mamy całkiem niezły wzór na obliczanie CSOD(n):

$$\mathrm{CSOD}(n) = -\frac{(\left[\frac{n}{2}\right]+2)(\left[\frac{n}{2}\right]-1)}{2} + n\left(\left[\frac{n}{k}\right] - 1\right) - \sum_{i=2}^{\left[\frac{n}{k}\right]} n \,\%\, i +$$

$$+ \sum_{l=k}^{3} \frac{(l-1)}{2} \left(\left[\frac{n}{l-1}\right] + \left[\frac{n}{l}\right] + 1\right)\left(\left[\frac{n}{l-1}\right] - \left[\frac{n}{l}\right]\right). \tag{4}$$

7 Dobór parametru k

Parametr k dobieramy tak, żeby zmniejszyć ilość działań. Mnie wyszło, że trzeba zminimalizować funkcję $\left[\frac{n}{k}\right] + 7k$ (dlaczego?). Czyli

$$k \approx \sqrt{\frac{n}{7}}$$

(czemu?). Ile działań będzie w najgorszym przypadku ($n = 2\,000\,000\,000$)?

8 Obliczenia według wzoru (4)

Odpowiedząc na wszystkie pytania, zadane wyżej, zorganizuj obliczanie według wzoru (4) i zalicz zadanie. Mój program działał 0:00.137 minut.

9 Psychologia

Czasami można przyśpieszyć program uwzględniając psychologię autora zadania. Z dużym prawdopodobieństwem pośród danych testowych będą przykłady z treści zadania oraz skrajne przypadki. W danym zadaniu można dodać na przykład taki **switch**:

```
switch(n){
case 1:
    cout<<"0\n"; break;
case 2:
    cout<<"0\n"; break;
case 100:
    cout<<"3150\n"; break;
case 200000000:
    cout<<"12898681201837053\n"; break;
case 2000000000:
    cout<<"1289868132101422932\n"; break;
case 1999999999:
    cout<<"1289868129103864851\n"; break;
default:
    //Właśnie obliczenia...
}
```

10 A new function

Mimo nazwy zadania, obliczana funkcja nie jest nową. Widziałem tę funkcję w książce [1], kiedy byłem dzieckiem.

Literatura

[1] Д. О. Шклярский, Н. Н. Ченцов, И. М. Яглом, *Избранные задачи и теоремы элементарной математики,* Москва, ,,Наука", **1977**.