

Wstęp do programowania w języku C

Grupa MSz w czwartki

Lista 5 na zajęcia 17.11.2022

Zadanie 1. (10 punktów na pierwszej pracowni, 5 punktów na drugiej)

Na wejściu dana jest liczba naturalna n oraz sekwencja n liczb całkowitych oddzielonych białymi znakami. Podsekwencję otrzymujemy przez wybór niektórych z tych n liczb; powtarzające się liczby traktujemy oddzielnie, tak samo jak różne. Napisz program obliczający liczbę wszystkich podsekwencji, w których suma liczb wnosi 0 i które zawierają nie więcej niż M liczb.

Założenia:

1. $1 \leq n \leq 36$.
2. Liczby są z przedziału $[-10^{12}, 10^{12}]$.
3. $1 \leq M \leq 10$.

Program powinien działać w sensownym czasie dla wszystkich dopuszczalnych danych. Oczywiście trzeba pamiętać o optymalizacji.

Wskazówka: Trzeba sprawdzić wszystkie możliwe podzbiory i najłatwiej to zrobić rekurencyjnie.

- Przykład 1 ($M = 10$):

5

1 2 5 -3 -2

Wynik:

4

Wyjaśnienie: 1+2-3, 2-2, 5-3-2 i pusta podsekwencja.

- Przykład 2 ($M = 10$):

12

1 1 1 1 1 1 1 1 1 1 -9 -10

Wynik:

11

- Przykład 3 ($M = 10$):

5

1000000000000 -999999999999 4 -3 2

Wynik:

2

- Test wydajnościowy ($M = 10$) (łatwy):

30

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
999999999999 -195 -145

- Test wydajnościowy ($M = 10$) (trudny):

36

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
999999999999 -8888888888 8888888888 -8888888888 -465 -225

Zadanie 2. (10 punktów)

Rozwiąż zadanie 1 z innymi założeniami:

1. $1 \leq n \leq 1000$.
2. Liczby są z przedziału $[-10^3, 10^3]$.
3. $M \leq 20$.

Tutaj wynik może być dużą liczbą, ale wystarczy podać go modulo 2^{64} .

- Generator testu wydajnościowego:

```
printf("1000\n");  
for (int i = 0; i < 500; i++) printf(" 1000 -1000");
```

Wersja łatwiejsza za 7 punktów: Bez ograniczenia M na wielkość podzbioru.

Zadanie 3. *Przypominam, że do każdej listy w SKOSie jest jeszcze do zrobienia zadanie dla sprawdzaczki, które ma osobny termin.*