Omówienie zadania `Wycinek`

Filip Konieczny

4 lipca 2024

1 Omówienie

Mamy daną tablicę $(T_i)_{i=1}^n$ liczb całkowitych i chcemy znaleźć jej najdłuższy spójny fragment, którego elementy sumują się do s. Gdyby liczby podane na wejściu były nieujemne, wtedy zadziałałoby rozwiązanie oparte na gąsienicy: trzymamy dwa indeksy i sumę elementów między nimi. Jeśli suma elementów jest mniejsza badź równa s, to przesuwamy przedni indeks, jeśli za duża to tylni. Jeśli mamy sumę dokładnie s to porównujemy aktualny przedział z najlepszym.

Ponieważ oba wskaźniki przesuniemy sumarycznie liniowo wiele razy, podejście to ma złożoność liniową. Niestety, w naszej sytuacji nie działa ono (istnieje wiele kontrprzykładów, zachęcam do znalezienia jakiegoś), bo powiększanie przedziału niekoniecznie zwiększa sumę elementów w nim.

Oznaczmy $pre_0 = 0, pre_i = T_1 + T_2 + \ldots + T_i$, tzn. tablicę sum prefiksowych. Zauważmy, że przedział (i, j) ma sumę s wtedy i tylko wtedy gdy $pre_j - pre_{i-1} = s$ i $j \geq i$. Będziemy kolejno rozważać wszystkie j-ty i znajdywać dla danego indeksu najlepsze (tzn. najmniejsze) pasujące i. Jak to zrobimy?

Dla ustalonego j chcemy znaleźć najmniejsze i spełniające $pre_{i-1} = pre_j - s$. Jak to osiągnąć? Wszystkie napotkane do tej pory sumy prefiksowe trzymamy w mapie (hashmapie) M, takiej, że $M[pre_{i-1}] = i$. Wtedy, najlepsze i dla danego j to $M[pre_j - s]$ (o ile istnieje).

Robimy tak dla każdego j i aktualizujemy mapę odpowiednio (uwaga: nie należy nadpisywać pola w mapie, jeśli już ma jakąś wartość - szukamy najmniejszego indeksu o danej sumie). Rozwiązanie ma złożoność $\mathcal{O}(n \log n)$ lub oczekiwaną $\mathcal{O}(n)$ w zależności od tego czy użyjemy `std::map` czy `std::unordered_map`.