



Problema MinimizeSum

Clasa a X-a

Autor Stud. Alex-Nicolae Pop, Universitatea Babeș-Bolyai, Cluj-Napoca

Soluție în $\mathcal{O}(N^K)$

Se încearcă toate posibilitățile de a aplica *operația magică*.

Soluție în $\mathcal{O}(NK)$

Se poate aplica o strategie de tip greedy în care la fiecare pas se alege poziția care scade cel mai mult suma șirului și se aplică *operația magică* pe ea. Algoritmul se termină atunci când nu mai sunt poziții care ar putea optimiza suma.

Soluție în $\mathcal{O}(N^2)$

Se optimizează soluția precedentă cu observația că este optimă orice poziție care micșorează suma șirului.

Soluții în $\mathcal{O}(K)$, $\mathcal{O}(K \log K)$

Se poate menține un `std::unordered_set` sau un `std::set` cu pozițiile care ar putea îmbunătăți suma. Ca în soluția precedentă, nu contează care din poziții este aleasă.

Soluții în $\mathcal{O}(N \log N)$

Pentru rezolvarea în întregime a problemei trebuie făcută următoarea observație legată de *operația magică*:

- Considerăm șirul de diferențe d definit prin $d_i = v_i - v_{i-1}$.
- În urma aplicării *operației magice* pe poziția i , efectul asupra șirului d este de interschimbare a valorilor d_i cu d_{i+1} .

Putem reduce problema la a o rezolva pe șirul d folosind faptul că $v_i = v_1 + d_2 + \dots + d_i$, deci suma șirului v este $nv_1 + (n-1)d_2 + (n-2)d_3 + \dots + 2d_{n-1} + d_n$. Cum valoarea v_1 nu se modifică în urma aplicării operației, trebuie să minimizăm suma $(n-1)d_2 + (n-2)d_3 + \dots + 2d_{n-1} + d_n$.

Pentru a minimiza șirul, dorim să avem valorile mai mici la început, iar valorile mai mari la final. Așadar, pentru soluția optimă putem sorta șirul de diferențe d_2, d_3, \dots . Se poate demonstra că această soluție este optimă folosind tehnica de „Exchange arguments”.

Astfel, numărul minim de operații pentru a sorta șirul folosind doar interschimbări de elemente consecutive este egal cu numărul de inversiuni din șir.

Această strategie demonstrează și greedy-urile din soluțiile precedente deoarece dacă o anumită poziție scade suma, această operație rezolvă o inversiune din șirul d .

Pentru a calcula numărul de inversiuni rapid, se poate folosi oricare metodă clasică precum Merge Sort sau folosind un arbore indexat binar sau arbore de intervale.