

Tabăra de pregătire a lotului național de informatică

Deva, 20 aprilie – 27 aprilie 2013

Baraj 3

Sursa : `rotatii.c`, `rotatii.cpp`



Problema 3 – rotații

100 puncte

Stud. Adrian Budău – Universitatea din București

Observatie:

Fie S șirul sumelor parțiale. Dacă $S[N] < 0$ atunci nu există soluție (afisam -1), altfel gasim i astfel incat $S[i] \leq S[j]$ oricare ar fi j între 1 și N si asta e solutia. (Daca $i = N$ afisam 0). Se poate demonstra asta urmarind ce se intampla la sirul dublat si facandu-i sumele parțiale.

Soluția 1 – 20 puncte

Aplicand observatia, vom avea o solutie in $O(n*m)$.

Solutia 2 – 50 puncte

Neavand operatia de **Delete K**, putem tine un arbore de intervale in care tinem minimul sumelor parțiale curente. Procesam operatiile in ordinea inversa a citirii , problema rezolvandu-se asemanator cu schi de pe Infoarena.

Solutia 3 – 70 puncte

Impartim sirul initial in bucati de \sqrt{N} in care tinem minimul sumelor parțiale si valorile pe fiecare bucata. Atunci cand intalnim un **Query** rezolvarea minimului pe vecturl Sum se face in \sqrt{N} . Problema apare atunci cand in urma inserarilor sau stergerilor bucata se dubleaza sau ramane goala, atunci reconstruim fiecare bucata in $O(N)$ total.

Solutia 4 – 80 de puncte

Se aplica **Solutia 3** in caz ca intalnim operatii de tipul **Delete K** . Altfel **Solutia 2** .

Solutia 5 – 100 de puncte

Tinem un treap in care mentinem sirul. Operatiile de adaugare si sergere ale unui element sunt clar in $\log N$. Pentru a rezolva si query-ul in $\log N$ (adica sa cautam cea mai mica suma partiala) vom tine pentru fiecare nod si urmatoarele informatii: Suma elementelor din subarbore si cea mai mica suma partiala din subarbore(independent de ce e in exterior). Acesta informatii pot fi updatate in $O(1)$ cand modificam fii unui nod deci inserarea si stergerea raman $O(\log N)$. Pentru query doar vom cobora in stanga sau in dreapta pe treap pentru a cauta suma partiala de cost minim.