

BARAJ 1 – Descrierea soluțiilor

Sortare prin inversiuni

O sugestie valoroasă ascunsă în textul problemei este să considerați întâi numai șiruri de 0 și 1. Pentru acest caz, se poate găsi un algoritm cu costul $O(N \lg N)$, bazat de divide-and-conquer. Se împarte șirul în două bucăți de mărimi egale. Fiecare bucată se sortează recursiv. Apoi șirul va avea forma: 0..01..1 0..01..1

Acum se poate efectua o singură inversiune în jurul punctului unde am împărțit șirul. Astfel, inversăm elementele 1 din prima jumătate și elementele 0 din a doua jumătate, și obținem un șir sortat. Costul acestei inversiuni este cel mult numărul de elemente din șirul pe care îl sortăm. Deci pe fiecare nivel de recursivitate avem un cost de $O(N)$, și în total costul este $O(N \lg N)$.

Pentru cazul general, folosim algoritmul anterior ca pas auxiliar pentru a implementa QuickSort, și obținem un cost de $O(N \lg^2 N)$. Pentru a sorta cu QuickSort, alegem mediana șirului, și partiționăm șirul, punând elementele mai mici ca mediana în stânga și elementele mai mari ca mediana în dreapta. Cum se poate face această partiționare eficient? Să presupunem ca toate elementele mai mici ca mediana le renumerotăm ca „0”, și toate elementele mai mari ca mediana le renumerotăm ca „1”. Acum putem aplica algoritmul pentru cazul 0-1, și cu un cost de $O(N \lg N)$ putem face partiționarea. Apoi continuăm sortarea generală pe bucata din stânga și cea din dreapta. Fiindcă costul de partiționare e $O(N \lg N)$, spre deosebire de $O(N)$ la un QuickSort normal, avem un cost total de $O(N \lg^2 N)$.

Constanta se poate calcula fără probleme și se vede ușor că e mai mică decât $\frac{1}{2}$. Algoritmul are un cost mai mic de 2 milioane pentru cel mai dificil caz.

Trans

Problema admite o soluție simplă, având complexitatea $O(N \cdot K)$ pentru fiecare tip de camion. Se calculează pentru fiecare bloc de piatră i , suma totală minimă plătită, $SMIN[i]$, astfel încât fie transportate primele i blocuri. Pentru aceasta, se variază j între i și $i-K+1$, reprezentând primul bloc din cadrul transportului care conține blocul i . Această soluție poate fi optimizată până la o complexitate liniară. Se va menține o structură de tip coadă, sortată crescător. Fiecare element al structurii corespunde cazului în care ultimul transport efectuat începe la una din pozițiile $1, 2, \dots, i$. Pe măsură ce se calculează valorile $SMIN[i]$, se introduc aceste valori (puțin modificate), la sfârșitul cozii, eliminându-se elementele de la sfârșit care au valori mai mari decât valoarea nou introdusă (sfârșitul cozii funcționează ca o stivă). Pentru a calcula $SMIN[i]$, se alege valoarea de la începutul cozii. Dacă această valoare corespunde unei poziții de start la distanță mai mare decât K față de i , atunci valoarea se elimină și se încearcă utilizarea valorii următoare. Complexitatea finală a soluției este $O(N \cdot Q)$.

Peri

Soluția simplă (de 40 de puncte) are complexitatea $O(N^4)$ și nu vom insista asupra ei. Există o soluție mai bună, cu complexitate $O(N^3)$. Se alege linia de sus și linia de jos a dreptunghiului ($O(N^2)$ posibilități). Pentru fiecare pereche avem o „fâșie” și trebuie să căutăm soluția cea mai bună din fâșia respectivă. Pentru aceasta vom parcurge coloană cu coloană, iar la fiecare pas vom ține soluția cea mai bună din stânga (și numărul soluțiilor cele mai bune). O soluție din stânga constă în 3 părți ale perimetrului dreptunghiului (linia verticală din stânga și cele două linii orizontale). La fiecare pas adăugăm la soluția curentă diferența dintre numărul de 1 și 0 de la două poziții ale matricii (una care completează linia de sus și una pentru cea de jos) și comparăm cu o nouă soluție posibilă ce ar corespunde dreptunghiurilor care au marginea din stânga la coloana curentă. Acest pas constă în $O(N)$ operații, rezultând complexitatea totală de $O(N^3)$.