BARAJ 2 – Descrierea soluţiilor

SALA

Vom demonstra prin inducţie matematică cu pasul 3 că numărul maxim de băieţi în sală este egal cu $x_n = \left[\frac{n(n+1)+1}{3}\right]$.

Pentru n=1, 2, 3 verificările sunt imediate.
Să considerăm afirmaţia adevărată pentru n şi să o demonstrăm pentru n+3.
Fie $a_1, a_2, \ldots, a_n, a_{n+1}, a_{n+2}, a_{n+3}$ o aşezare pentru n+3 persoane. Următoarele două rânduri sunt date prin vectorii;
$b_1, b_2, \ldots, b_n, b_{n+1}, b_{n+2}$
$c_1, c_2, \ldots, c_n, c_{n+1}$

Pentru aceste trei rânduri de persoane numărul minim de fete este n+2 şi deci numărul maxim de băieţi este 2n+4.
Astfel obţinem $x_{n+3} \le 2n+4+x_n$.

Conform ipotezei de inducţie se obţine că

$x_{n+3} \le 2n+4+[\frac{n(n+1)+1}{3}] = [\frac{(n+3)(n+4)+1}{3}]$. Adică tocmai ce trebuia demonstrat.

Valoarea maximă $x_n = \left[\frac{n(n+1)+1}{3}\right]$ se poate obţine pentru următoarea aşezare în sală:

```
bbfbbfbbf…
fbbfbbfb…
bfbbfbb…
...
b
```

Ghizi

Se observa ca problema devine o problema de flux, prin urmatoarea transformare:

- momentele intregi de timp se transforma in nodurile unui graf ;
- un interval [T1,T2) asociat unui ghid introduce un arc de la T1 la T2, cu capacitatea 1. Doua noduri pot fi conectate prin mai multe arce.

Se observa ca un drum de la sursa (nodul 0) la destinatie (nodul 100) acopera fiecare moment de timp din [0,100) exact o data. Astfel, problema poate fi reformulata :
"Dandu-se graful construit conform transformarii de mai sus, sa se determine K drumuri, disjuncte din punct de vedere al muchiilor, de la sursa la destinatie"
Problema este in mod evident o problema de flux ; se introduce un arc de capacitate K de la o sursa fictiva la nodul 0 si se executa un algoritm de flux maximal.

ajutor

Solutie O(N*M)

Pentru fiecare punct dintre cele M se calculeaza distanta pana la fiecare din cele N puncte si se ia cea mai mica valoare

Solutie O(N*N+MlogN)

Se considera intai cele N puncte si se face o preprocesare. Se iau coordonatele x intr-un vector. Se sorteaza. Acelasi lucru si cu y. Fiecare punct de o coordonata din vectorul x, iar alta din vectorul y va reprezenta un nod intr-o retea ortogonala (grid). Pentru aceste noduri se calculeaza printr-o dinamica de complexitate O(N^2) distanta pana la cel mai apropiat post de prim ajutor. Mai exact acest lucru se face :

Se fac 4 dinamici. Una incepand din stanga-sus, alta incepand din stanga-jos, alta incepand din dreapta-jos, alta incepand din dreapta-sus. Acestea se fac cu ajutorul unei matrice a[i,j] reprezentand distanta pana la cel mai apropiat post de prim ajutor, daca am merge numai in stanga si sus (de exemplu pentru prima matrice), din pozitia (i,j) in grid, adica al i-a coordonata y, dupa sortare si a j-a coordonata x dupa sortare. Calculul se face O(N^2), pentru ca valoarea lui a[i,j] nu depinde decat de vecinii din stanga si sus.

Dintre toate cele 4 valori se retine minimul si se trece intr-o matrice.

Considerente de memorie pentru compilatoarele Borland :

Nu avem nevoie decat de o singura matrice, cea finala. Ea incape pe word (respectiv unsigned int) (pentru ca distanta maxima e de 64000) si ocupa : 400*400*2=320000 octeti, adica arhisuficient.

Pentru cele 4 dinamici nu este nevoie decat de 2 vectori : nivelul curent si nivelul anterior.

Fiecare dintre cele M puncte se rezolva (se afla minimul) in O(logN), pentru ca nu trebuie decat sa aflam in ce celula din grid se afla punctul in care ne afla (2 cautari binare in vectorii x si y), iar acolo sa vedem in care dintre cele 4 colturi e mai bine sa mergem.

Solutie O(N*N+M)

Din cauza limitarilor coordonatelor pentru puncte (pentru a putea fi rezolvabila si in Borland), s-a nascut si posibilitatea de a raspunde la o cerere in O(1). Acest lucru se face prin preprocesare pe cei doi vectori x si y. Se mai retin 2 vectori in care pentru fiecare coordonata x se afla coloana i si pentru fiecare coordonata y se afla linia j a gridului obtinut prin preprocesare.

Pentru Borland mai este nevoie de 2 vectori de 32000 de elemente de tip word (respectiv unsigned int), adica : 2*32000*2 = 128000.

In total memoria folosita este de mai putin de 448 K, mai mult decat suficienta.

Trebuie subliniat faptul ca acesta solutie e mult mai usor de implementat si mai rapida, insa nu exista decat pentru limitarile curente ale coordonatelor. Solutia precedenta merge pentru numere oricat de mari, chiar reale…

Solutie Sprite

Presupunand ca o parte din teste erau random (cum este si cazul la 4 teste mari), se putea imparti toata zona de 32000x32000, in zone de 3200 pe 3200 adica 100 de zone (sau in zone mai mici, dar mai multe). Pentru fiecare punct al nostru de locatie este suficient (doar pe testele random) sa ne uitam doar la punctele ce se afla in jurul punctului nostru, adica in cele 8 zone. Se luau aproximativ 60 de puncte.