

Problema 3 –spider

Autor:

Budai István, Liceul Teoretic "Nagy Mózes"

Soluție 35 pct

Se definesc două matrice: **A** și **B** cu maxim **1000** de rânduri și maxim **1000** de coloane, precum și două tablouri, **v** și **w**, având cel mult **1000000** de elemente. În matricea **A** se citesc valorile înălțimilor clădirilor (numere naturale din intervalul **[1, 10000000]**). Se construiește câte unui drum, pornind de la toate elementele matricei **A**, și se reține drumul de lungime maximă.

Se utilizează o funcție care primește, prin intermediul parametrului **pas**, numărul de ordine al drumului care se construiește. Funcția întoarce înălțimea clădirii cu diferența de înălțime minimă, și care respectă condițiile din enunț. Funcția verifică în matricea **B** dacă clădirea de coordonate **(i, j)** a mai fost - sau nu - vizitată de Spiderman: dacă **B[i][j]=pas**, atunci clădirea de coordonate **(i, j)** a fost deja vizitată pe parcursul drumului cu numărul de ordine **pas**, deci clădirea nu este eligibilă. Dacă **B[i][j]** este diferit de **pas**, atunci se alege clădirea respectivă și lui **B[i][j]** i se atribuie valoarea **pas**.

Se utilizează un subprogram care primește, prin parametrii de intrare **r** și **c**, coordonatele unei clădiri și construiește drumul care poate fi parcurs pornind de la aceea clădire. Pentru selectarea unei clădiri se utilizează funcția descrisă mai sus. Dacă drumul cu numărul de ordine actual este mai lung decât precedentul drum, se va reține lungimea drumului în variabila **k**, precum și elementele vectorului de direcții **v** în vectorul **w**, care va avea lungimea **k** și va reține direcțiile corespunzătoare drumului maxim. Elementele vectorului **v** se definesc astfel:

v[i]=1, dacă direcția aleasă pentru săritura **i** este **Nord**;

v[i]=2, dacă direcția aleasă pentru săritura **i** este **Est**;

v[i]=3, dacă direcția aleasă pentru săritura **i** este **Sud**;

v[i]=4, dacă direcția aleasă pentru săritura **i** este **Vest**.

Programul principal va apela acest subprogram pentru toate elementele matricei **A**, pornind de la elementul de coordonate **(1, 1)**, parcurgând matricea de la linia **1** la linia **m**, și elementele pe linii, de la **1** la **n**. Datele unui drum se vor reține doar dacă lungimea sa este **strict** mai mare decât cea mai mare lungime obținută până în acel moment. La scrierea datelor în fișierul de ieșire (**k** și coordonatele clădirilor care formează drumul maxim) se va porni de la coordonatele primei clădiri, iar coordonatele clădirilor care formează drumul maxim se vor determina prin interpretarea corectă a elementelor vectorului **w**.

Complexitate: $O(N*N*M*M)$

Soluție 100 pct

Pornind de la soluția anterioară, observăm că, dacă nu am avea două valori identice pe poziții alăturate, nu are sens să calculăm din fiecare poziție de început care este lungimea maximă, deoarece putem stoca într-o matrice auxiliară **best[i][j]** lungimea maximă a unei secvențe care începe pe poziția **(i, j)**, iar atunci când determinăm un drum să actualizăm toate aceste costuri ale pozițiilor prin care trece.

Cum putem avea două valori vecine cu aceeași valoare, acest tablou **best** trebuie schimbat puțin în **best[x][y][d]** având semnificația: lungimea maximă a unei secvențe care începe pe poziția **(x, y)** care nu face prima săritură în direcția **d** (**d** poate avea valori de la 0 la 3, fiecare valoare semnificând una din cele 4 direcții posibile).

Complexitate: $O(N*M)$