drum - Descrierea soluţiei

Autor Carmen Mincă

O soluţie se poate obţine prin aplicarea metoda programării dinamice, metoda înainte. Se utilizează două masive tridimensionale `A` şi `T`. În masivul `A`, `A[1][1][1]` va memora valoarea asociată punctului de coordonate `(1,1,1)`,..., `A[k][i][j]` va memora valoarea asociată punctului de coordonate `(k,i,j)`, pentru $1 \le k \le n$, $1 \le i \le k$, $1 \le j \le k$. Masivul T se va iniţializa cu valorile din masivul A şi va fi utilizat pentru a reţine valorile sumele maxime parţiale calculate pentru drumurile construite.

Pentru un punct de coordonate `(k,i,j)`, suma maximă care se poate calcula, pentru un drum care porneşte din acest punct, se obţine pe baza relaţiilor:

`T[n][i][j]=A[n][i][j]` , pentru $1 \le i \le n$, $1 \le j \le n$. (k=n)

`T[k][i][j]=A[k][i][j]+max{T[k+1][i][j+1],T[k+1][i+1][j],T[k+1][i+1][j+1]}`
pentru `k=n-1,n-2,...,1`, $1 \le i \le k$, $1 \le j \le k$.

Suma maximă va fi reţinută în `T[1][1][1]`. Pentru a afişa numerelor punctelor de pe un drum de sumă maximă, refacem "traseul" prin care a fost obţinută suma maximă, pornind de la `T[1][1][1]` sau se memorează coordonatele punctelor în timpul construirii sumelor parţiale maxime.

Pentru exemplul din enunţ, se pot reprezenta punctele separat, pe plane:

Planul de ecuaţie: k=1

P_1(1,1,1)

Planul de ecuaţie: k=2

P_2(2,1,1)	P_3(2,1,2)
P_4(2,2,1)	P_5(2,2,2)

Planul de ecuaţie: k=3

P_6(3,1,1)	P_7(3,1,2)	P_8(3,1,3)
P_9(3,2,1)	P_{10}(3,2,2)	P_{11}(3,2,3)
P_{12}(3,3,1)	P_{13}(3,3,2)	P_{14}(3,3,3)

Valorile asociate punctelor:

Planul de ecuaţie: k=1

3

Planul de ecuaţie: k=2

6	5
7	2

Planul de ecuaţie: k=3

4	5	8
7	6	1
7	8	13

Se obţin două drumuri: D_1=(P_1, P_4, P_{13}) şi D_2=(P_1, P_5, P_{14}) şi $D_1 < D_2$. Astfel drumul căutat este D_1=(P_1, P_4, P_{13}) iar suma maximă este S=3+7+8=3+2+13=18.

atac - Descrierea soluţiei

Asociem hartii un graf neorientat cu n vârfuri şi m muchii, fiecărui oraş corespuzându-i un vârf şi fiecărui drum direct corespunzându-i o muchie. Notam cu k numarul de noduri care sunt legate printr-o muchie de nodul X. Fie aceste noduri X1, X2, ..., Xk.

Din fiecare nod Ui care contine o unitate, vom face o cautare in latime pentru a afla lungimile minime ale drumurilor de la acest nod la toate nodurile Xj, 1<=j<=k.

Solutia 1. (100 puncte) - Tiberiu Daneţ

Problema se reduce la a realiza un cuplaj maxim de cost minim in graful bipartit in care una dintre multimi este formata din nodurile ce contin unitati, iar cealalta multime de nodurile X1, X2, ..., Xk (fiecare unitate trebuie dusa intr-un nod Xj si nu are rost sa ducem mai multe unitati in acelasi nod Xj). Costul de pe muchia ce leaga nodul Ui de nodul Xj este lungimea minima a drumului de la Ui la Xj gasita anterior.

Complexitatea cuplajului maxim de cost minim in graful bipartit complet cu k + U noduri se face in complexitate O(k * k * k * U) (un drum alternant de crestere se poate obtine cu algoritmul Bellman-Ford si sunt k drumuri alternante de crestere).

Solutia 2. (60 puncte) - Alin Burţa

Daca numarul unitatilor disponibile este suficient de mic (pentru 60% din teste avem u<=8), determinarea modului in care cuplam cele u unitati cu cele k noduri vecine cu X se poate realiza cu un algoritm de tip backtracking.

virus - Descrierea soluţiei

Autor Pătcaş Csaba

Problema se poate rezolva prin mai multe metode, având complexităţi diferite.

Fie S suma lungimilor viruşilor.

O abordare bazată pe algoritmul naiv de căutare al subşirului are complexitatea $O(S \cdot m)$¸ şi poate obţine 10 de puncte.

Construirea unui arbore de sufixe se poate face în $O(m^2)$, după care numărarea apariţiei fiecărui şir se poate face în O(S). Deci aceasă solutie are în total complexitatea $O(m^2+S)$ şi obţine 30 de puncte.

Aplicarea de n ori a unui algoritm liniar de potrivire a şirurilor (de exemplu Knuth-Morris-Pratt, sau Boyer-Moore) are complexitatea $O(n \cdot m+S)$ şi obţine între 30 şi 60 de puncte.

Construirea şirului de sufixe al şirului de biţi şi se poate face prin diferiţi algoritmi cu complexităţi de $O(m \cdot \log^2 m)$, $O(m \cdot \log m)$ sau $O(m)$. După obţinerea şirului de sufixe, numărul de apariţii al fiecărui virus se poate determina prin două căautări binare.

Această abordare are în total complexitatea $O(m \cdot \log^2 m+S \cdot \log m)$, $O(m \cdot \log m + S \cdot \log m)$ sau $O(m+S \cdot \log m)$ şi poate obţine 70-80 de puncte.

Pentru obţinerea punctajului maxim este nevoie de o implementare eficientă a algoritmului Aho-Corasick.