BARAJ 2 – Descrierea soluţiilor

Sea (Radu Berinde)

Soluţia evidentă la această problemă are complexitatea **O(MNlgN)** şi se realizează prin **M** sortări în funcţie de distanţă. Acest tip de soluţie obţine în jur de 30 de puncte. Soluţia următoare ar fi folosirea unui algoritm de selecţie pentru a obţine complexitatea probabilistică de **O(MN)** şi 60 de puncte.

Să ne concentrăm însă asupra altei soluţii. Pentru fiecare far (în ordinea sortată după **x**) ţinem un vector cu vapoarele care au **x**-ul mai mic decât **x**-ul farului curent. Pentru fiecare far se încearcă o resortare a listei – însă nu de la 0. Având lista deja sortată putem încerca interschimbarea unor elemente consecutive (bubble-sort). Se observă că de-a lungul întregului algoritm vom interschimba aceleaşi două vapoare maxim o dată. Astfel complexitatea devine **O(N(N+M))**.

Putem să optimizăm soluţia aceasta: în loc să verificăm la fiecare far ce interschimbări au apărut, putem „prezice" momentele (ca **x**) unde apar interschimbările. Pentru aceasta vom ţine o structură cu **N-1** valori – valoarea **i** este coordonata **x** unde apare interschimbarea între elementele **i** şi **i+1** ale vectorului. Cazul în care această valoare există este cel în care un vapor se află în stânga altui vapor, dar are coordonata **y** mai mică (în caz contrar valoarea este considerată infinit). Punctul în care se schimbă ordinea celor două este coordonata **x** a centrului cercului determinat de cele două vapoare şi de restricţia coordonatei **y** a centrului, care trebuie să fie **0**. Structura menţionată trebuie să suporte aflarea minimului şi modificarea unei valori, şi poate fi implementată folosind un heap sau un arbore de selecţie. Astfel avem o secvenţă (fără să fie neapărat explicită) de evenimente: adăugarea unui vapor sau interschimbarea a două vapoare din vector. La fiecare eveniment recalculăm valorile din structură pentru elementele din vector care se modifică. Pe parcursul rezolvării evenimentelor, calculăm şi informaţia cerută pentru fiecare far (fiind sortate de la stânga la dreapta se pot citi pe parcurs din fişier). Există maxim **N(N-1)/2** evenimente de interschimbare şi maxim **N** de adăugare a unui vapor, deci complexitatea finală este **O(N^2lgN + M)**.

Poliţie (Em. Cerchez)

Soluţia 1 (Radu Berinde): Vom sorta evenimentele după timpul la care ele se termină. Vom ţine o matrice **A[i][j]** – costul maxim care se poate obţine astfel încât primul poliţist dă amenzi numai dintre primele **i** evenimente, iar al doilea dintre primele **j**. Pentru calculul eficient al acestei matrici, se poate precalcula un vector **Last[i]** – ultima amendă mai mică ca **i** care nu se suprapune cu **i**. Încadrarea amenzilor de tip 3 în soluţie are loc numai la calculul poziţiilor din matrice de pe diagonala principală. Recurenţele sunt:

Când **i < j**: **A[i][j] = A[i][j-1],** dacă evenimentul **j** este de tip 3 sau

max(A[i][j], A[i][Last[j]] + valoarea amenzii), dacă evenimentul **j** nu este de tip 3

Când **i = j**: **A[i][i] = max(A[i-1][i-1], A[Last[i]][Last[i]] + val amenzii),** dacă evenim **i** este de tip 3 sau

max(A[i-1][j-1], A[Last[i]][i-1] + val amenzii). dacă evenimentul **j** nu este de tip 3

Când **i > j**: **A[i][j] = A[j][i]**

Complexitate: **O(N^2)**.

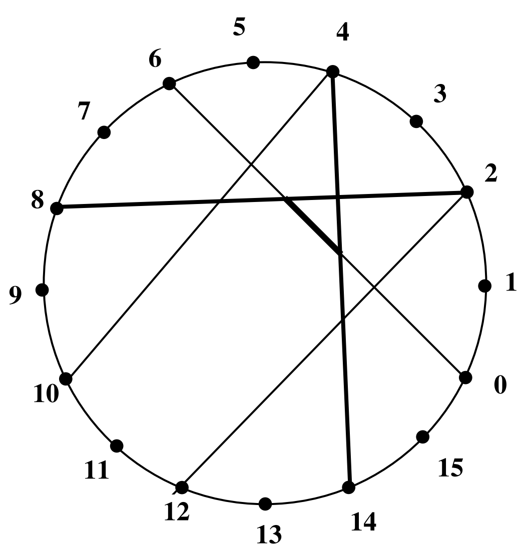
Soluţia 2 (EmCerchez):

Fie smax[i][t1][t2]=suma maximă care o pot încasa cei doi politişti
cu amenzile i, i+1, ... , N în conditiile în care poliţiştul 1 este disponibil la momentul t1, iar poliţistul 2 este disponibil la momentul t2.
Dacă amenda i este de tip 1 sau 2, atunci pentru a calcula smax[i][t1][t2] vom alege varianta optimă din următoarele două cazuri:

a. primul poliţist dă amenda i, dacă este posibil (t1<=timpul de a plicare a amenzii i)
b. al doilea poliţist dă amenda i, dacă este posibil (t2<=timpul de aplicare a amenzii i)

Să notăm cu tip tipul amenzii:
/*primul politist da amenda */
Dacă (t1<=t) atunci
  Dacă (smax[i][t1][t2]< S[tip]+smax[i+1][t+T[tip]][t2])
    smax[i][t1][t2]=S[tip]+smax[i+1][t+T[tip]][t2];
/*al doilea politist da amenda */
Dacă (t2<=t) atunci
    Dacă (smax[i][t1][t2]< S[tip]+smax[i+1][t1][t+T[tip]])
      smax[i][t1][t2]=S[tip]+smax[i+1][t1][t+T[tip]];
Dacă evenimentul este de tip 3 ambii poliţişti dau amenda dacă este posibil:
Dacă (t1<=t && t2<=t) atunci
    Dacă (smax[i][t1][t2]< S[3]+smax[i+1][t+T[3]][t+T[3]]) atunci
      smax[i][t1][t2]=S[3]+smax[i+1][t+T[3]][t+T[3]];
Observăm că pentru a calcula optimul pentru amenzile i, i+1, ..., n este necesar să cunoaştem valorile optime pentru amenzile i+1, ..., n.
Prin urmare, nu este necesar să utilizăm un tablou tridimensional, sunt suficiente două matrice de dimensiune T*T.
Deci, complexitate spaţiu T*T, complexitate timp T*T*N.

Poligon (Rodica Pintea)



Se calculează c=cmmdc(n,k) şi numărul de laturi L=n/c.
Pentru determinarea ariei poligonului colorat se poate determina fie latura, fie raza cercului circumscris. Se analizează intersecţia unei coarde duse de Geo (de exemplu prima) cu alte coarde duse ulterior. O codificare analitică a punctelor de pe cerc poate fi Pi (r cos(i∗u), r sin(i∗u)), unde $u=\frac{2\pi}{n}$.
Dacă n este divizibil cu k, poligonul regulat obţinut are raza egală cu raza cercului dat. Altfel, se calculează ca intersecţie a două coarde care au două capete consecutive (0,k) şi (c,c+k).
Dacă n este divizibil cu k, poligonul regulat obţinut are latura egală cu coarda care subîntinde c arce. Altfel, se calculează latura=distanţa dintre intersecţiile coardelor (c,c+k) şi respectiv (k+c, n-c) cu coarda (0,k). Figura analizată pentru n=16 şi k=6 este cea alăturată.