



Paul Diac, Facultatea de Informatica Iasi

Soluție - problema 1 – piscina

100 puncte

În explicație vom considera doar soluțiile care au în comun punctul $(0, 0)$. Celelalte 3 posibilități se pot obține similiar iar cel mai simplu în implementare este să oglindim toate punctele. De exemplu x_i devine $(N - x_i)$ pentru toate punctele, similiar și pentru y_i ($M - y_i$), iar la final atât pentru x_i cât și pentru y_i . Răspunsul va fi valoarea maximă dintre cele 4 soluții obținute prin oglindire.

O primă observație este faptul că piscina de arie maximă va avea câte un copac pe fiecare din acele două laturi care nu sunt adiacente punctului $(0, 0)$. Altfel, ar putea fi extinsă iar aria ar crește. O singură excepție de la această regulă ar fi atunci când și una din aceste laturi este comună cu una din laturile dreptunghiului.

Putem considera oricare două puncte pentru a limita dimensiunea piscinei în partea de dreapta și sus. Dimensiunea piscinei limitate de aceste două puncte i și j este $(\max(x_i, x_j) * \max(y_i, y_j))$. O astfel de piscină este validă doar dacă nu există alte puncte în acest dreptunghi.

O(P³) – 30 puncte se pot obține dacă iterăm toate perechile (i, j) de puncte care pot limita piscina și apoi iar toate celelalte puncte pentru a testa dacă ele sunt sau nu în interiorul piscinei.

O(P²) – 60 puncte se pot obține optimizând astfel: limităm dimensiunea piscinei doar pe coordonata y prin orice punct (x_i, y_i) . Latura maximă pe dimensiunea x va fi acum egală cu x -ul minim al oricărui punct care are valoare y mai mică decât y_i . (punctele cu y mai mare oricum sunt în afara piscinei datorită limitei stabilite). Astfel nu trebuie iterate decât două for-uri imbricate, de unde rezultă complexitatea.

O(PlogP) – 100 puncte se pot obține dacă îmbunătățim ideea cu încă un pas. Sortăm punctele după y crescător, și considerăm în această ordine că fiecare punct limitează dimensiunea y a piscinei. Valoarea maximă x până la care putem extinde piscina este posibil doar să scadă pe măsura ce iterăm punctele, atunci când un nou punct are valoare x mai mică decât punctul anterior (el va avea oricum valoare y mai mare pentru că punctele sunt sortate). Dacă noul punct are valoare x mai mare, el va fi ignorat, deoarece are valoare atât x cât și y mai mare și deci nu poate fi un punct limită optim (piscina conține colțul $(0, 0)$).

Pentru a trata simplu cazul în care piscina e limitată de dimensiunile N și respectiv M se pot adăuga punctele $(0, M)$ și $(N, 0)$, apoi algoritmul nu trebuie să trateze cazul diferit.

Este necesară folosirea tipului **long long** pentru a obține punctajul maxim. Folosind doar numere pe 32 biți, se pierd cel puțin câte 10, 20 și respectiv 30 puncte pentru fiecare soluție în ordine.

Soluție 2-Divide et impera

DI - soluția inițială Divide & Impera fără nici o modificare

DI_o1 - soluția cu optimizarea o1 (un if)

DI_o2 - soluția cu optimizarea o2 (alt if)

DI_o21 - soluția cu ambele optimizări / if-uri:

DI - 40 puncte

DI_o1 - 50 puncte

DI_o2 - 60 puncte

DI_o21 - 70 puncte