



Descrierea soluției - roboți

Autor: prof. Sofia Vițelaru
C. N. „Frații Buzesti” Craiova

Cerința 1. 50 puncte

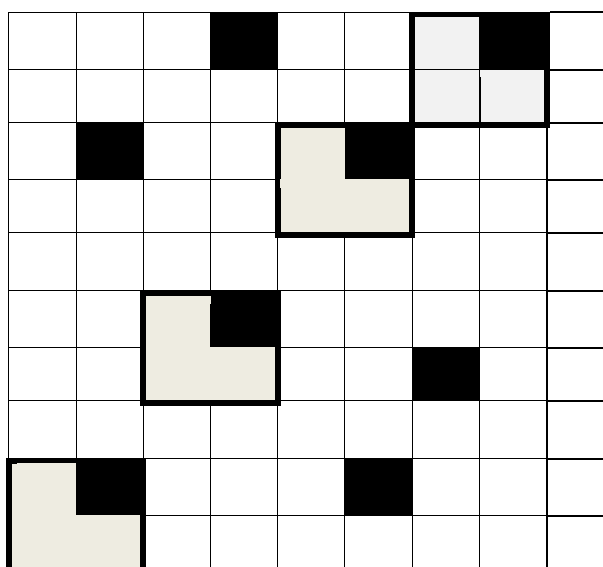
prof. Constantin Gălățan,
C. N. „Liviu Rebreanu” Bistrita

Soluția 1 – Complexitate $O(n * T * \log T)$

Se aplica o tehnică greedy. Se sortează coordonatele copacilor crescător după linie și pentru aceeași linie, crescător după coloană. Se plasează robotul curent cu colțul dreapta sus pe primul copac (cel mai de sus și din stânga). Din acest moment se simulează mișcarea. Se parcurg punctele în ordinea sortată. Pentru fiecare copac întâlnit pot exista situațiile: este acoperit de robot (copac tip A), se află în dreapta și mai jos față de colțul dreapta sus al robotului (copac de tip B), se află sub latura de jos a robotului (copac de tip C) sau se află la stânga robotului (copac de tip D). Copacii de tip A și de tip C se șterg din șirul de copaci, copacii de tip D se pierd și va trebui să fie tăiați de următorii roboți, iar în cazul în care apare un copac de tip D, robotul se mută cu colțul stânga sus pe acel robot și îl va tăia. Când robotul curent atinge colțul stânga jos al terenului, următorul robot va parcurge șirul de copaci rămași în șir. Algoritmul se oprește când toți copacii din șir au fost tăiați (șterși din șir)

Soluția 2

Din desenul de mai jos se observă că numărul total de roboți necesari tăierii tuturor copacilor este egal cu lungimea maximă a unei secvențe de copaci în cazul din exemplu patru) care sunt plasați descrescător după linie și la aceeași linie crescător după coloană, astfel încât distanța pe orizontală între doi roboți alăturați în secvență să fie mai mare sau egală decât latura robotului. Toți ceilalți roboți de pe teren pot fi tăiați de unul dintre acești roboți.



Se parcurg copacii de sus în jos și de la dreapta la stânga.

În funcție de metoda aleasă pentru determinarea celei mai lungi secvențe de copaci care respectă condiția menționată mai sus se pot obține diverse complexități (de exemplu, $O(n * n)$ cu programare dinamică respectiv $O(T * \log^2 n)$ cu aib 2d).



Cerința 2. 50 de puncte

Se caută binar latura minimă a roboților care trebuie folosiți pentru a nu depăși numărul admis de roboți utilizați. Numărul de roboți folosiți se face la fel ca la cerința 1. Pentru un algoritm de tip greedy, complexitatea soluției la cerința 2 este $O(Q \cdot n \cdot T \cdot \log T)$

Președinte:
Radu Eugen Boriga

Vicepreședinte subcomisie clasa a IX-a
Constantin Gălățan