



PROBLEMA 2 – tromino

Autor: Vlad STOIAN

Se pot face cateva observatii:

- La fiecare pas, patrutul de latura 2^K se poate imparti in 4 patrute (cadrane) de latura 2^{K-1}
- Daca punctele se afla in acelasi cadran, rezolvam aceeași problema pentru un $K-1$
- Daca punctele se afla in cadrane diferite, drumul va trece prin mijlocul patrutului

Pornind de la aceste observatii, putem rezolva problema in 2 moduri: recursiv si iterativ.

Varianta recursiva simpla - 60 pct

- Daca punctele se afla in acelasi cadran, rezolvam recursiv pt $K-1$
- Daca punctele se afla in cadrane diferite, calculam recursiv:
 - drumul de la primul punct la mijloc in $K-1$
 - drumul de la al doilea punct la mijloc in $K-1$
 - costul de a trece dintr-un cadran in altul (1 atunci cand cadranele sunt alaturate, 2 atunci cand cadranele sunt opuse)

Varianta recursiva optimizata (60 - 100 pct)

Solutia de mai sus poate fi optimizata folosind memoizare pentru a nu calcula de multe ori aceeași distanta. (~ 60 pct)
O alta optimizare este folosirea formulei explicata mai jos, atunci cand punctele se afla pe diagonala. (~ 100 pct)

Varianta iterativa - 100 pct

Observam ca daca ambele puncte se afla pe diagonala principala (sau secundara), numarul de mutari se poate calcula printr-o formula: $|S_r - F_r| + |S_c - F_c|$. Astfel, pentru orice K , pentru a calcula distanta de la orice punct, la punctul din dreapta jos, facem urmatoorii pasi:

- Daca punctul se afla in primul cadran, rezolvam pentru $K-1$
- Daca se afla in unul dintre celelalte cadrane, incercam sa translatam punctul in primul cadran tinand minte un cost de translatare. Acest cost va reprezenta cati pasi in plus adauga translatarea la drumul curent, si va trebui scazut mai tarziu. Rezolvam pentru $K-1$

Dupa ce va fi translatat de K ori, punctul va ajunge in $(0,0)$. Deci raspunsul va fi: distanta dintre $(0,0)$ si $(2^{K-1}, 2^{K-1})$ minus suma tuturor costurilor de translatare.