

castel – descrierea soluției

În ciuda aparentei simplități a problemei, rezolvarea nu este tocmai trivială. Principala problemă se datorează faptului că, în momentul în care printesa deschide o cameră și ia cheia aflată acolo, ea poate să se întoarcă și să deschidă camere pe care anterior ar fi “vrut” să le viziteze, dar nu a putut. O parcurgere simplă care nu ține cont de această posibilitate ar primi 10-20 de puncte.

O altă posibilitate este un BF modificat astfel încât la fiecare pas să se verifice dacă se mai poate deschide vreuna din camerele adiacente cu fiecare dintre nodurile deja procesate. Această soluție are complexitatea $O(N^2 * M^2)$ și obține aproximativ 50 de puncte.

O îmbunătățire constă în menținerea unei liste cu elementele camerele nevizitate care sunt vecine ale celor deschise deja, verificarea de mai sus făcându-se doar pentru aceasta. Soluția de 100 de puncte are complexitatea $O(N*M)$ și se bazează pe următorul algoritm:

Q – lista de noduri vizitate și neprocesate (preferabil implementată ca o coadă)

Lst[i] = lista camerelor pe care le pot deschide după ce am luat cheia din camera #i

A[i] = camera care trebuie să fi fost deschisă pentru a deschide camera #i

U[i] = 1, camera #i a fost vizitată

0, altfel

Q = {K}

cat timp Q ≠ ∅

extrage x din Q

pentru toate elementele z din Lst[x]

 u[z] = 1

 Q = Q ∪ {z}

pentru toți vecinii y ai lui x cu U[y] = 0

daca U[A[y]] = 1

 u[y] = 1

 Q = Q ∪ {y}

altfel

 Lst[A[y]] = Lst[A[y]] ∪ {y}

Este posibilă existența unor soluții alternative având aceeași complexitate.