

## Solutia 1

Pentru a putea iesi din labirint trebuie sa gasim un drum de la A la B si un drum de la B la C. Putem deci incerca sa gasim un drum de la B la A si un drum de la B la C.

Acest lucru se realizeaza printr-o parcurgere cu algoritmul lui Lee, considerand ca punct de pornire fiecare B. Ne oprim cand algoritmul atinge un A si un C. Astfel, vom memora cel mai scurt drum catre A si cel mai scurt drum catre C.

Rezultatul este suma celor doua drumuri -1. (-1 pentru a nu contoriza de doua ori B -ul de pornire)

Complexitate finala:  $O(\text{numar\_de\_B\_uri} * n * m)$

## Solutia 2

Pentru a putea gasi iesirea din labirint trebuie sa parcurgem un drum de la A la B si in continuare la C. Acest lucru ne poate duce la ideea de a imparti problema in parcurgerea de la A la B si parcurgerea de la B la C.

Pentru parcurgerea de la A la B cel mai scurt drum se afla cu algoritmul lui Lee, considerand ca puncte de pornire toate casutele de tip A. Algoritmul functioneaza si cu mai multe puncte de pornire. Corectitudinea este data de faptul ca se proceseaza secvential punctele. Nu se va procesa o casuta cu timpul  $k+1$  inaintea unei casute cu timpul  $k$ . Daca o casuta este procesata ( am trecut prin casuta ) timpul din aceasta casuta este cel minim. Se poate ajunge in casuta (din nou) cu un timp asemanator sau mai mare. Parcurgere de la casute de tip A la casute de tip B se face prin marcarea timpului in tabel (tabel1) si introducerea casutei in coada. Ordinea de procesare a punctelor de pornire nu conteaza, se obtine acelasi timp de parcurgere.

Pentru parcurgerea de la B la C trebuie sa luam in considerare ca ajungem la punctele B la timpi diferiti si ca putem parcurge din nou zone deja vizitate din labirint. Pentru a pastra corectitudinea algoritmului nu trebuie sa prelucram o casuta cu timpul  $k+1$  inaintea uneia cu timpul  $k$ .

Pentru a respecta aceste lucruri parcurgerea de la B la C se va realiza prin marcarea timpului intr-un alt tabel (notat tabel2). Avem nevoie de un alt tabel pentru a putea parcurge din nou labirintul. Pentru a respecta timpii de procesare a casutelor din tabel2 acestea se vor introduce **in aceeasi coada** (coada folosita pentru primul Lee). Prin folosirea unei singure cozi se garanteaza ca nu se va procesa o casuta cu timpul  $k+1$  inainte uneia cu timpul  $k$ , indiferent daca este din tabel1 sau tabel2.

### algoritm

1. tabel1, tabel2 contin matricea bordata si pregatita de parcurgere (sunt identice).

2. se citesc toate casutele A si se marcheaza in tabel1 cu 1 (se ajunge in ele la timpul 1)

se introduc in coada pozitia casutelor A si numarul tabelii (1)

3. cat timp sunt elemente in coada

3.1 se extrage din coada - casuta si numarul tabelii

3.2 daca elementul extras este din tabel1

daca casuta are vecini nevizitati (in tabel1)

se marcheaza vecinul in tabel1 si se introduc in coada (vecinul, nr. tabelii (1))

daca casuta este de tip B

se marcheaza in tabel2 (cu acelasi timp ca in tabela 1) si se introduce in coada (vecinul, nr. tabelii (2))

3.3 daca elementul extras este din tabel2

daca casuta are vecini nevizitati (in tabel2)

se marcheaza in tabel2 si se introduce in coada (vecinul, nr. tabelii (2))

daca casuta este C - raspuns gasit (valoarea din tabel2 din casuta C)

Complexitate finala:  $O(n*m)$