

# EscapeLight

*Autori: stud. Moca Andrei-Cătălin,  
stud. Lorintz Alexandru, stud. Măierean Mircea, Universitatea Babeș-Bolyai Cluj-Napoca*

## Descrierea problemei

Problema presupune parcurgerea unui labirint reprezentat printr-o matrice de dimensiuni  $N \times M$ , unde fiecare celulă (cameră) poate fi de unul din patru tipuri:

- **Tipul 0:** cameră cu becul stins,
- **Tipul 1:** cameră cu becul aprins,
- **Tipul 2:** cameră fără bec,
- **Tipul 3:** cameră echipată cu un întrerupător.

## Observații

Problema devine interesantă datorită posibilității de a schimba dinamic starea iluminării în labirint. La accesarea unei camere de tip 3 se poate acționa un întrerupător care inversează starea becurilor din camerele conectate (aprins devine stins și invers). Din acest motiv, pe lângă poziția în labirint, este necesar să ținem evidența configurației întrerupătoarelor (adică, decizia de a le apăsa sau nu). Deoarece numărul de camere cu întrerupător este restrâns ( $P \leq 10$ ), se poate utiliza o reprezentare prin mască de biți, fiecare bit indicând dacă întrerupătorul a fost acționat sau nu.

Starea sistemului este, astfel, reprezentată printr-un triplet ce conține:

- Poziția curentă în matrice;
- Configurația curentă a întrerupătoarelor.

## Subtask 1 (10 puncte: $K = 0$ )

În acest caz, nu există nicio conexiune între întrerupătoare și becuri, astfel încât starea de iluminare a fiecărei camere rămâne fixă pe tot parcursul parcurgerii. Problema se reduce la găsirea unui drum minim printr-un grid unde se poate trece doar prin camerele luminoase inițiale. Se poate aplica o parcurgere în lățime (BFS) clasică, fără a fi necesară actualizarea dinamică a stării.

## Subtask 2 (10 puncte: $K = 1$ )

Aici există o singură conexiune între un întrerupător și bec. Deși modificările stării de iluminare se pot produce, efectul este izolat la un singur toggle. Strategia constă în extinderea spațiului de căutare pentru a include, pe lângă poziția din labirint, și posibilitatea de a acționa sau nu întrerupătorul din camera curentă.



**Subtask 3 (10 puncte:  $P = 1, K \geq 2$ )**

În acest subtask se lucrează cu un singur întrerupător, astfel încât configurația este reprezentată de un singur bit. Chiar dacă întrerupătorul poate influența starea mai multor camere, evaluarea finală a fiecărei camere se reduce la determinarea parității toggle-ului aplicat.

**Subtask 4 (15 puncte:  $P = 2$ )**

În acest caz, existența a două întrerupătoare extinde spațiul de stare la  $N \times M \times 2^2$ , adică patru configurații posibile. Fiecare cameră de tip 0 sau 1 își are starea finală determinată de efectul combinat al toggle-urilor de la cele două întrerupătoare. Strategia rămâne aceeași: parcurgerea în lățime care, pentru fiecare poziție, explorează posibilitatea de a trece în camerele adiacente și, la intrarea în camerele cu întrerupător, opțiunea de a modifica configurația curentă.

**Restul testelor**

Pentru determinarea drumului minim se efectuează o căutare extinsă în graful de stări, unde fiecare stare este definită prin poziția în labirint și configurația întrerupătoarelor. Se poate utiliza algoritmul de BFS 0-1, destinat grafurilor cu muchii de costuri 0 și 1, folosind un deque.

- **Tranzițiile:** Se consideră deplasarea către cele 4 direcții (sus, jos, stânga, dreapta). Pentru fiecare deplasare, se verifică dacă camera următoare este luminoasă conform configurației curente, mutând cu cost 1. În plus, dacă se intră într-o cameră cu întrerupător, se poate opta pentru a modifica configurația (toggle), generând astfel o nouă stare, cu cost de deplasare 0.
- **Evitarea repetării:** Se păstrează o evidență a stărilor deja vizitate (combinație poziție și configurație) pentru a asigura că fiecare stare este procesată o singură dată.

Această abordare permite explorarea întregului spațiu de stări de dimensiune  $O(N \cdot M \cdot 2^P)$ .