

# Lights Out

---

V našom zadaní máme implementované riešenie hry Lights Out pomocou 3 vyhľadávacích algoritmov.

## Reprezentácia stavu hry

---

Keďže v nasledujúcich algoritmoch sa vyžaduje vyhľadávanie v stavovom priestore (grafe), pokladali sme za potrebné definovať si štruktúru Node, ktorá obsahuje:

- Stav hry (Rozpoloženie zasvietených políčok)
- Odkaz na rodiča (Stav, z ktorého sme dostali aktuálny stav)
- Akciu, ktorá bola vykonaná na rodičovskom stave
- $h(n)$  - Hodnotu heuristickej funkcie
- $g(n)$  - Hodnotu cenovej funkcie (vzdialenosť od počiatočného stavu)

## DFS

---

Algoritmus DFS máme implementovaný iteratívne, zásobník je reprezentovaný pomocou dátovej štruktúry **stack**.

DFS spočíva v brute-force prístupe, ktorý prechádza stavový priestor (graf) nasledovným spôsobom:

1. Vyberie si posledne pridaný stav zo zásobníka
2. Ak stav ešte nebol expandovaný, expanduje ho a označí ako už expandovaný
  - Ak susedný stav, ktorý algoritmus našiel expandovaním predstavuje finálny stav hry, algoritmus našiel riešenie

## Greedy

---

Algoritmus Greedy máme implementovaný iteratívne, zásobník je reprezentovaný pomocou dátovej štruktúry **priority queue**, ktorú zoradíme na základe heuristickej funkcie  $h(n)$ .

Heuristická funkcia vyzerá nasledovne:

$$h(n) = \text{Pocet zasvietenych policok}$$

Algoritmus prechádza stavový priestor (graf) nasledovným spôsobom:

1. Vyberie stav s minimálnou hodnotou  $h(n)$  zo zásobníka
2. Ak stav predstavuje finálny stav hry, algoritmus našiel riešenie
3. Ak stav ešte nebol expandovaný, expanduje ho a označí ako už expandovaný

## A\*

---

Algoritmus A\* máme implementovaný iteratívne, zásobník je reprezentovaný pomocou dátovej štruktúry **priority queue**, ktorú zoradíme na základe funkcie  $f(n)$ .

Funkcia  $f(n)$  vyzerá nasledovne:

$$f(n) = h(n) + g(n)$$

$$f(n) = \text{Pocet zasvietenych policok} + \text{vzdialenost od pociatocneho stavu}$$

Algoritmus prechádza stavový priestor (graf) nasledovným spôsobom:

1. Vyberie stav s minimálnou hodnotou  $f(n)$  zo zásobníka
2. Ak stav predstavuje finálny stav hry, algoritmus našiel riešenie
3. Ak stav ešte nebol expandovaný, expanduje ho a označí ako už expandovaný