

# Šachy

## Řešení problému

Úkolem je efektivně zodpovědět kolik nepřátel je Frodo schopen porazit, aniž by byl objeven, ale pokud by byl objeven ihned napsat „Byl jsem odhalen“.

## Hledání počet zabitelných nepřátel

### Validace kroků

Vytvoříme si pravidla pro každého nepřítele, pro  $z$ , uložíme 2 pozice, pro  $c$  uložíme obecnou rovnici pro danou přímku, pro  $t$  uložíme 8 pozic, pro  $v$  uložíme 2 obecné rovnice.

Pokud budeme chtít zkontrolovat, zda můžeme jít na dané políčko projdeme uloženými pravidly, dokud nenajdeme jedno, které se nachází se v pravidlech a tam nemůžeme, ale pokud se nic nenajde, můžeme tam jít.

### Hledání

Nejdříve se podíváme, zda je naše startovní pozice validní. Pomocí  $A^*$  budeme procházet vždy všechny kombinace nepřátel jako cíle našeho pathfindingu s validováním kroků (slouží jako stěna) rekurzivně, ale pokud zjistíme že už nemůžeme nikoho porazit podíváme se jaký je největší počet poražených nepřátel a pokud jsme porazili více, uložíme nové skóre.

Po každém zabití odebereme pravidla, která byla vytvořena tím specifickým nepřítelem. Tyto pravidla nejsou uložena jen jedna, při hledání cest každá kombinace má uložená svoje pravidla, protože je to rekurzivní funkce.

Nejlepší skóre uložíme také pokud se nám podaří porazit všechny nepřátele, v tom případě končíme hledání dalších možností a vracíme nejlepší skóre, ale pokud je nejlepší skóre 0 vrátit „Byl jsem odhalen“.

## Zdůvodnění řešení

Toto řešení bude vždy správné, protože projde všechna pravděpodobná řešení co nejefektivněji a vrátí největší počet poražených nepřátel.

## Zdůvodnění nejlepšího řešení

Toto řešení je nejlepší řešení, protože bychom neměli mít možnost najít řešení bez procházení možných cest, já je ale procházím efektivně bez potřeby udržování mapy.

## Časová komplexita

Toto řešení nejdříve parsuje vstup v čase  $O(n)$ . Poté rekurzivně prochází všechny pravděpodobné kombinace pořadí poražených nepřátel, což má časovou složitost  $O(Z * Z * (b^d))$ , kde  $b$  je průměrný počet úspěšných tahů na jeden tah a  $d$  je hloubka nejkratší cesty nalezené algoritmem.

## Paměťová komplexita

Toto řešení vytváří 2x slovníky, pro pravidla a lokace nepřátel, pro každou instanci pathfindingu s časovou složitostí  $O(k \cdot 2n)$ , kde  $k$  je počet projetych cest při pathfindingu.