

Cvičení z Algoritmizace a Programování 1

11. cvičení

Algorithmizace

Prohledávání – obecný iterativní algoritmus verze 1

```
objevene = {pocatecni_stav}
k_prozkoumani = (pocatecni_stav)
while k_prozkoumani:
    stav = k_prozkoumani.get()
    for naslednik in naslednici(stav):
        if naslednik not in objevene:
            objevene.add(naslednik)
            k_prozkoumani.add(naslednik)
```

Prohledávání – obecný iterativní algoritmus verze 2

```
prozkoumane = {}  
k_prozkoumani = (pocatecni_stav)  
while k_prozkoumani:  
    stav = k_prozkoumani.get()  
    if stav not in prozkoumane:  
        prozkoumane.add(stav)  
        for naslednik in naslednici(stav):  
            k_prozkoumani.add(naslednik)
```

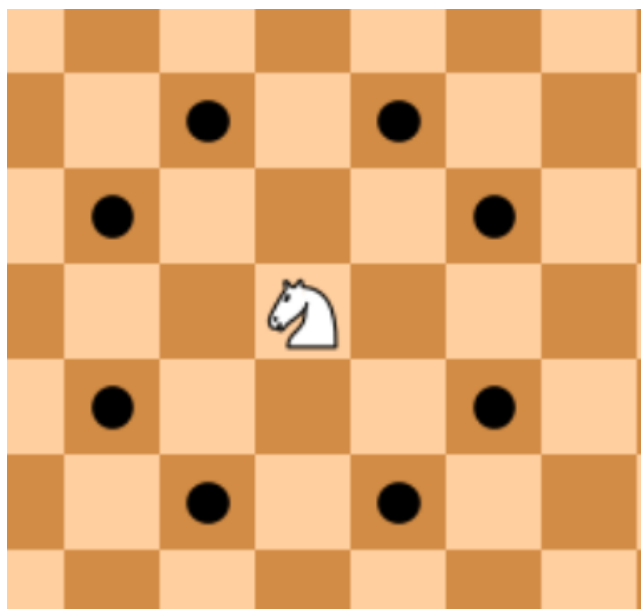
Řešení DÚ

- Přelévání vody
- Šestka

Hledání nejkratší cesty

- Šachovnice 8x8
- Najděte nejkratší cestu (na počet tahů) jezdcem z políčka A na políčko B

- Tahy jezdce:



Kružnice v grafu

- Neorientovaný graf
- Orientovaný graf

Programování

Poslední domácí úkol – Cesta králem

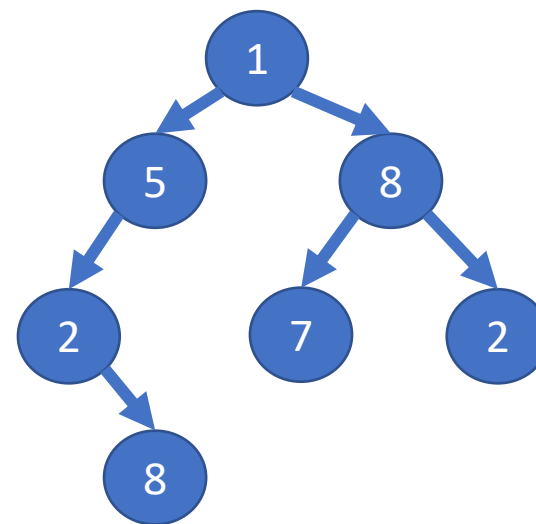
- <https://recodex.mff.cuni.cz/app/assignment/a6cc7605-65f5-4224-a32e-5ac9e0882cc9>

Rekurze – zefektivnění

- Fibonacciho čísla – viz `fibonacci.py`

Binární stromy

- Maximum
 - Rekurzivně
 - Nerekurzivně
 - „Nejmělcí“ (nejbližší kořeni)



Načítání stromu ze souboru – nestihlo se

- Soubor na webu (strom.txt)
- Formát:
 - Každý vrchol má unikátní identifikátor (typu string) a hodnotu (int)
 - 1. řádek: identifikátor kořene
 - Další řádky:
 - 1 řádek = 1 vrchol stromu, v libovolném pořadí!
 - Formát řádku:
 - `id -> hodnota, id_levého_dítěte, id_pravého_dítěte`
 - id dětí mohou být prázdná (= vrchol dané dítě nemá)
 - Např. `n1 -> 5, ,n2` – vrchol „n1“ má hodnotu 5, levé dítě nemá, jeho pravé dítě je vrchol „n2“