KSI 2014/2015

Úloha 5-2: Stará hra

Jan Horáček
Gymnázium, Brno, Vídeňská 47; jan.horacek@seznam.cz

30. dubna 2015

1 Analýza problému

Po chvíli zamyšlení zjišťujeme, že stěžejní částí řešení bude nalezení největšího souvislého podgrafu grafu *svet*. Tento podgraf reprezentuje souvislé území jednoho hráče. Pokud takový podgraf nalezneme, jméno jeho vlastníka je jménem vítěze. Vydejme se tedy takový podgraf hledat.

2 Řešení problému

2.1 Slovník *hraci*

Zaveďme nejprve slovník *hraci* indexovaný hráči (konkrétně např. jejich jmény, čísly, ...), který každému hráči přiřadí jednu `uint` hodnotu – velikost jeho největšího území. Před začátkem algoritmu pro hledání největších podgrafů vytvoříme v tomto slovníku záznam pro všechny hráče a velikost jejich území položíme rovnu 0.

2.2 Hledání největšího podgrafu

Hledání největšího podgrafu grafu *svet*, který patří konkrétnímu hráči *hrac* provedeme lineárním prohledáváním všech podgrafů. Pro tyto účely obohatíme každý vrchol informací o tom, zda-li jsme ho už navštívili. Mimo *uzemi.v* a *uzemi.sousedi* bude tedy existovat i *uzemi.navstiveno*, které bude zpočátku inicializováno na *false* pro všechna území v celém grafu *svet*.

Algoritmus bude lineárně (každý vrchol pouze jednou) procházet všechny nenavštívené vrcholy, pro každý takový vrchol provede DFS, které bude expandovat všechny vrcholy s totožným majitelem *hrac* a které spočte velikost celého podgrafu. Pokud bude tato velikost větší, než *hraci*[*hrac*], uloží do *hraci*[*hrac*] velikost nového největšího území.

Algoritmus můžeme zapsat:

```
1 for vrchol in svet: vrchol.navstiveno = false
2
3 for vrchol in svet:
4         if not vrchol.navstiveno:
5                 velikost = DFS(vrchol)
6                 if velikost > hraci[vrchol.hrac]:
7                         hraci[vrchol.hrac] = velikost
```

```
1 def DFS(vrchol):
2         velikost = vrchol.v
3         vrchol.navstiveno = true
4         for soused in vrchol.sousedi:
5                 if (soused.hrac == vrchol.hrac) and (not
                    soused.navstiveno):
6                         velikost += DFS(soused)
7         return velikost
```

Po skončení tohoto algoritmu (který běží lineárně vzhledem k počtu vrcholů) dostaneme velikosti největších území všech hráčů.

3 Nalezení hráče s největším územím

Nalezení hráče s největším územím je pak otázka nalezení maxima ve slovníku *hraci*, resp. nalezení všech maxim. To umíme v lineárním čase a nejhůře v lineární paměti (pro případ, kdy by měli všichni hráči stejnou velikost území) vzhledem k velikosti slovníku.

4 Závěr

Časová náročnost celého algoritmus je tedy $O(|V|+|E|+|hraci|)$, kde $|V|$ je celkový počet vrcholů grafu (celkový počet území na mapě), $|E|$ je celkový počet hran grafu (celkový počet hranic mezi územími), a $|hraci|$ je počet hráčů.

Paměťová složitost algoritmu je $O(|V|+|E|+|hraci|)$, protože potřebujeme uložit do paměti celý graf *svet* a slovník *hraci*, ve kterém si udržujeme velikosti největších území hráčů.

Tento algoritmus je nezávislý na počtu hráčů, tedy fungoval by pro libovolný počet hráčů. Pro umožnění neutrálních území stačí přidat podmínku pro to, aby neutrální území nebyla nikdy expandována (na řádek 4 do prvního kódu).