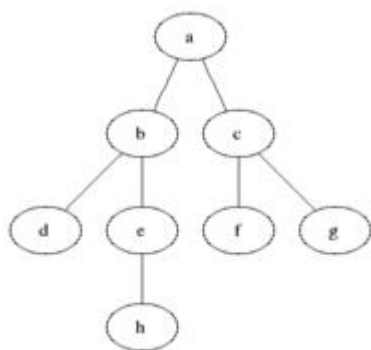


Křižovatky 2.kolo Fiks 2021/2022

Kryštof Olík

Zformulujme si zadání této úlohy. Na vstupu dostaneme počet křižovatek M , počet hran křižovatek N (které si také můžeme představit jako jednosměrné cesty mezi křižovatkami) a na konec počet otázek O . Následně máme N hran, které vedou jednosměrně z křižovatky A do křižovatky B . Nakonec je nám kladeno O otázek, které se nás ptají, zda se dokážeme dostat z křižovatky A do křižovatky B .

Nejprve si uložíme všechny hrany a otázky do vektorů. Poté si z uložených hran vytvoříme graf. Vytvoříme si list a do indexu pramenů připojíme všechny destinace hran. Tento graf má prostorovou složitost $O(M + N)$, kde M je počet křižovatek a N počet hran.



Pro vizualizaci můžeme vidět tento graf v obrázku nalevo. Na vrchu se nachází startovní křižovatka z otázky a a pod ní se někde nachází konečná křižovatka.

Začneme iterovat přes všechny dané otázky a zjistíme, zda se z křižovatky A dokážeme dostat do křižovatky B . Algoritmus začne prohledávat do šířky náš graf, dokud nenajde naší destinaci. To znamená, že postupujeme systematicky graf přes všechny křižovatkami. Nepoužíváme při svém prohledávání žádnou heuristickou analýzu. Pouze procházíme všechny křižovatkami a pro každou projdeme jejich všechny následovníky. Přitom si poznamenáváme předchůdce jednotlivých křižovatek.

Veškerí následovníci křižovatek získaný expandující křižovatkou jsou vkládány do FIFO fronty. FIFO fronta znamená, že první křižovatka, která do fronty vstoupila jí také první opustí. Všechny křižovatkami přes které jdeme označíme, že už byli objeveny, abychom přes ně už znovu nešli.

Jakmile najdeme cílovou křižovatku tak vložíme do výstupu, že cesta existuje. Pokud projdeme všemi možnými křižovatkami a cílovou křižovatkou nenajdeme, tak vložíme do výstupu, že cesta neexistuje. Poté uděláme ty samé kroky pro ostatní otázky. Tento celý algoritmus včetně cyklu otázek má časovou složitost $O(QM)$, kde Q je počet otázek a M je počet křižovatek.

Ještě musíme dokázat, že náš algoritmus funguje. Prohledávání do šířky zajistí, že projdeme přes všechny možné křižovatkami, tudíž se nemůže stát, že bychom nějakou cestu nenašli. Zároveň je algoritmus konečný, jelikož si pamatuje přes jaké křižovatkami již šel, tudíž se nemůže do nekonečna zacyklit.

K řešení připojuji algoritmus naprogramovaný v jazyce C++. Program samotný je přesným přepisem výše uvedeného algoritmu.

Algoritmus, který používáš, se jmenuje DFS. Ve skutečnosti ale existuje lepší algoritmus na tuto situaci (Kosarajův algoritmus). Tento algoritmus dokáže zkondenzovat celý graf na graf souvislých komponent, ve kterém je možné rychle odpovídat na dotazy. Při použití tohoto algoritmus je časová složitost $O(N+M+O)$ [-7]