Grafové algoritmy pro hledání nejkratší cesty

**1. Úvod**

Graf se skládá z vrcholů a hran. Používá se k modelování vztahů nebo propojení mezi objekty. Existuje několik typů grafů:

* **Neorientovaný graf**: hrany nemají směr (silnice)
* **Orientovaný graf**: každá hrana má směr (jednosměrka)
* **Ohodnocený graf**: hrany mají přiřazenou hodnotu (cena, vzdálenost či čas)

**Reprezentace grafu**

* **Matice sousednosti** – čtvercová matice, kde prvek na pozici (i, j) značí přítomnost a váhu hrany mezi vrcholy i a j
* **Seznam sousedů** – u každého vrcholu se ukládá seznam vrcholů, se kterými je spojen (efektivnější pro řídké grafy)

**Reálné využití grafů**

Grafy se používají v mnoha oblastech:

* Navigace (plánování trasy v mapových aplikacích)
* Počítačové sítě (směrování paketů)
* Plánování (např. rozvrh úloh, závislosti)

**2. Problém hledání nejkratší cesty**

Hledání nejkratší cesty znamená nalezení posloupnosti vrcholů, která spojuje dva body v grafu a minimalizuje součet vah hran, po kterých se pohybujeme.

**Záporné hrany**

Některé grafy mohou obsahovat hrany se zápornou váhou. Některé algoritmy (např. Dijkstrův) si s nimi neporadí správně. Pokud v grafu existuje záporný cyklus, nelze najít jednoznačnou nejkratší cestu – protože se po něm dá opakovaně „zkracovat“ cesta.

**3. Přehled algoritmů**

**a) Dijkstrův algoritmus**

* Patří mezi greedy algoritmy – vždy vybírá vrchol s aktuálně nejkratší známou vzdáleností
* Používá prioritní frontu pro efektivní výběr dalšího vrcholu
* Nevhodný pro záporné hrany, protože by mohl zvolit špatnou cestu
* Časová složitost:
  + S prioritní frontou: O((V + E) log V)
  + S jednoduchým polem: O(V²)

**b) Bellman-Fordův algoritmus**

* Funguje tak, že opakovaně uvolňuje hrany – zlepšuje odhady vzdáleností
* Provede až V−1 průchodů přes všechny hrany
* Funguje i s negativními hranami
* Umí detekovat záporný cyklus
* Časová složitost: O(V \* E)

**c) Floyd-Warshallův algoritmus**

* Vypočítá nejkratší cesty mezi všemi dvojicemi vrcholů
* Vhodný pro menší grafy nebo tam, kde potřebujeme kompletní přehled všech cest
* Časová složitost: O(V³)

**Závěr**

Grafové algoritmy pro hledání nejkratší cesty jsou důležitou součástí běžného života. Každý algoritmus má své výhody:

* **Dijkstra** je rychlý, ale jen pro nezáporné hrany,
* **Bellman-Ford** je univerzálnější, i když pomalejší,
* **Floyd-Warshall** se hodí pro kompletní výpočty mezi všemi vrcholy.