1. Kdy je Dijkstrův algoritmus ekvivalentní prohledávání do šířky?

   Pro graf s jednotkově ohodnocenými hranami.

2. Jak a pro jaké grafy se dá Dijkstrův algoritmus simulovat pomocí BFS? Jaká je asymptotická časová složitost výsledného algoritmu?

   Pro grafy s celočíselnými vahami hran to jde přímočaře podrozdělením hran (hranu váhy w nahradíme cestou délky w). Jednoduše jde tato konstrukce rozšířit na racionální váhy a i reálné váhy (vhodným zaokrouhlením, které nezmění vzájemné pořadí cest).

3. Mějme graf silniční sítě, ve kterém je pro každou silnici poznamenaná průměrná rychlost. Popište algoritmus, který najde mezi vrcholy u, v cestu maximalizující rychlost v nejpomalejším místě.

   Můžeme použít upravený Dijkstrův algoritmus:

   - Místo vzdálenosti budeme do vrcholů ukládat minimální rychlost na nejrychlejší cestě.
   - Budeme zavírat vrchol ne s *nejmenší* hodnotou (t.j. s nejkratší cestou), ale *největší* (s nejrychlejší cestou).
   - Hodnotu vrcholu přepíšeme pokud nová bude *větší*.
   - Při relaxaci u z v místo nastavení hodnoty u na *součet* ohodnocení hrany uv a hodnoty vrcholu v provedeme *minimum*.

4. Jak můžeme upravit Dijkstrův algoritmus, aby optimalizoval jiná kritéria místo délky cesty? Jaká kritéria lze takto použít?

   Kritérium si můžeme představit jako funkci, která hodnotí každou cestu: $h\colon V^* \to \mathbb{R}$. Standardní Dijkstrův alg. minimalizuje kritérium délky cesty. Kritérium můžeme i maximalizovat, jako v úloze 3.

   Z důkazu správnosti Dijkstrova algoritmu dostáváme tato omezení na kritérium:

   - Pro každou cestu $u_1, \ldots, u_k$ musí její prefix mít stejné nebo lepší ohodnocení (tedy $h(u_1, \ldots, u_{k-1}) < h(u_1, \ldots, u_k)$ pokud minimalizujeme) (kritérium je *monotónní*).

- Ohodnocení cesty závisí pouze na ohodnocení jejího prefixu bez poslední hrany a této jedné hraně.
- Každý prefix optimální cesty je také optimální (problém má *optimální podstrukturu*).

5. Můžeme zmírnit podmínky na kritéria použitím Bellmanova-Fordova algoritmu jako základu?

Bellmanův-Fordův algoritmus zjemňuje podmínku Dijkstrova algoritmu na absenci záporných hran (tedy každá hrana navíc cestu pouze zhorší) na podmínku absence záporných cyklů (tedy nemůžeme cestu donekonečna zlepšovat). Pro nás to znamená, že už nemáme podmínku monotonicity.

6. Pro množinu hodnotících kritérií je řešení *Pareto-optimální*, když neexistuje žádné jiné řešení, které by bylo v alespoň jednom kritériu ostře lepší a ve zbytku nebylo horší.
Upravte Dijkstrův algoritmus, aby pro množinu kritérií (splňujících podmínky z předchozího cvičení) hledal Pareto-optimální cesty. Jak se změní jeho časová složitost?

V tomto případě nám nestačí udržovat si v každém vrcholu jen jednu optimální cestu, kterou pak rozšiřujeme, protože taková neexistuje. Musíme udržovat množinu všech Pareto-optimálních cest (tzv. *Pareto front*) a zkusit vždy rozšířit každou z nich.

Algoritmus to násobně zpomalí, protože v každém vrcholu potřebujeme udržovat $\mathcal{O}\left(d^{k-1}\right)$ cest, kde k je počet kritérií a d je velikost jejich domény (počet možných ohodnocení).

7. Popište postup, kterým z výsledků Floydova-Warshallova algoritmu získáme posloupnost vrcholů, které cesta navštíví.

Pro hledání cesty z u do v nejprve najdeme nějaký její vnitřní vrchol w a pak rekurzivně zpracujeme cestu uw a wv: Vyzkoušíme všechny vrcholy $w \neq u, v$ a vybereme takový, kde vzdálenosti uw a wv se sečtou na vzdálenost uv (t.j. pro matici vzdáleností M platí $M_{uw} + M_{wv} = M_{uv}$).

8. Dokažte, že existuje množina nejkratších cest vedoucích z vrcholu v t.ž. tvoří dohromady strom.

Tento strom je generován Dijkstrovým algoritmem.

