Dynamické programování

1. Vyřešte problém nejdelší rostoucí podposloupnosti, ve které je povolen jeden pokles.

2. Na vstupu je vážený DAG G s jedním zdrojem sa jedním stokem t. Vymyslete algoritmus, který hledá nejkratší cestu z vrcholu s do vrcholu t.

3. Longest path (LPATH) (česky nejdelší cesta)
   Vstup: graf G, váhová funkce $w : E \to \mathbb{R}_0^+$.
   Výstup: cesta $P \subseteq V(G)$ taková, že $\sum_{v \in P} w(v)$ je minimální.

   Vyřešte problém LPATH na váženém stromě.
   Poznámka: problém LPATH je NP-těžký na obecných grafech, naštěstí ho řešíme jenom na stromě.

4. Weighted vertex cover (WVC) (česky vážené vrcholové pokrytí)
   Vstup: graf G, cenová funkce $w : V \to \mathbb{R}_0^+$.
   Výstup: co nejlevnější podmnožina vrcholů $S \subseteq V(G)$ taková, že pro $\forall e \in E(G) \exists v \in S : v \cap e \neq \emptyset$ a $\sum_{v \in S} w(v)$ je minimální.

   Vyřešte problém WVC na váženém stromě.
   Hint: zkuste to nejdříve na neváženém stromě.
   Poznámka: na obecných grafech je WVC NP-těžký (a dokonce i nevážená je NP-těžká), jde však o praktický problém a umíme jej „řešit“ nejrůznějšími způsoby. Například ho umíme 2-aproximovat, tzn. umíme najít řešení, které je nejhůře dvakrát horší než optimální řešení pro jakýkoliv vstup.

5. Je dán řetězec S. Vymyslete algoritmus, který v něm najde nejdelší podřetězec s, který je palindromem.

Minimální kostry

6. Ukažte, že algoritmy na hledání minimální kostry fungují i pro grafy s neunikátními vahami.

7. Máme vážený graf G a v něm jsme našli minimální kostru T. Nyní změníme váhu jedné hraně v grafu G, jak to ovlivní minimální kostru grafu G?

8. Navrhněte algoritmus, který hledá minimální kostru v grafu takovém, že váhy jsou celá čísla v rozmezí 1 až 5. Ideálně by měl pracovat v čase $O(n + m)$.

9. Jak realizovat Jarníkův algoritmus s haldou a s časovou složitostí $O(m \log(n))$?

Nejkratší cesty

10. Ukažte, jak pro libovolné n sestrojit graf na nejvýše n vrcholech takový, že mezi libovolnými dvěma vrcholy existuje $2^{\Omega(n)}$ nejkratších cest.

11. Je možné zbavit se záporných hran v algoritmech pro hledání nejkratší cesty tak, že ke všem hranám přičtu nějakou konstantu K?

12. Je dána mapa s městy a silnicemi mezi nimi. Každá cesta má nějakou délku a vybírá se na ní nějaké mýtné. Vymyslete algoritmus, který najde nejkratší cestu mezi dvěma zadanými městy takovou, že je zároveň nejlevnější.

13. Je dána mapa města, ve které jsou vyznačené mosty, které mají nějakou nosnost. Chceme převézt náklad z jednoho místa na druhé. Vymyslete algoritmus, který zjistí, jak nejtěžší náklad můžeme převézt, aby se po cestě žádný most nezhroutil.

14. V Tramtárii jezdí po železnici rychlíky, které nikde po cestě nestaví. V jízdním řádu je pro každý rychlík uvedeno počáteční a cílové nádraží, čas odjezdu a čas příjezdu. Nyní stojíme v čase t na nádraží A a chceme se co nejrychleji dostat na nádraží B. Navrhněte algoritmus, který najde takové spojení. Uvažujte, že časy jsou zadány v rámci jednoho dne s minutovou granularitou.