

CVIČENÍ 9

Téma: Nejkratší cesty

Cíle: Zvládnout techniku hledání nejkratších cest v grafu, získat praktické zkušenosti se základními algoritmy (Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson). Pracovat se složitostmi těchto algoritmů a zvážit vliv úprav nebo omezení na speciální typy grafů na jejich složitost.

Nejkratší cesty	
1.	Vytvořte obyčejný neorientovaný graf $G=\langle H,U \rangle : U= \{1, 2, \dots 7\}$, $H = \{[i,j] : 6 \leq i+j \leq 9\}$, tj. hrana mezi uzly i a j existuje právě tehdy, když $6 \leq i+j \leq 9$. Necht' pro ohodnocení hran platí $w(i,j) = i-j $. Určete vzdálenost z uzlu 7 do všech ostatních uzlů. Představuje minimální kostra tohoto grafu také strom vzdáleností z některého uzlu? Zopakujte úlohu pro případ, že hrana $[i,j]$ existuje právě tehdy, když $5 \leq i+j \leq 8$ a pro ohodnocení hran platí $w(i,j) = i+j$. (Je možné doplnit i další varianty.)
2.	Johnsonův algoritmus Na vhodném jednoduchém grafu se záporným ohodnocením hran (6 uzlů, 10 hran) ilustrujte použití Johnsonova algoritmu. Jak by dopadlo přehodnocení hran, kdyby původní ohodnocení všech hran bylo nezáporné?
3.	Bellman-Fordův algoritmus Na jednoduchém příkladu zopakujte princip Bellmanova-Fordova algoritmu.
4.	Floyd-Warshallův algoritmus Připomeňte strukturu Floyd-Warshallova algoritmu (promítněte/napište na tabuli jeho pseudokód). Diskutujte význam jednotlivých cyklů a dvojoperace uvnitř trojitého cyklu.
5.	Modifikace Dijkstrova algoritmu Necht' je v obyčejném neorientovaném grafu $G=\langle H,U \rangle$ zadáno ohodnocení w , které každé hraně $[u,v]$ přiřazuje reálné číslo $w(u,v)$ z intervalu $\langle 0,1 \rangle$. Toto hodnocení vyjadřuje spolehlivost komunikační linky $[u,v]$, tj. pravděpodobnost bezchybného přenosu po této hraně. Za předpokladu vzájemné nezávislosti těchto pravděpodobností navrhnete efektivní algoritmus určení nejspolehlivější komunikační cesty mezi dvěma zadanými uzly.
6.	Modifikace Bellman-Fordova algoritmu Upravte Bellmanův-Fordův algoritmus tak, aby nastavil hodnoty $d[u] = -\infty$ všem uzlům u , pro které existuje cesta z uzlu s do uzlu u procházející cyklem se zápornou w -délkou.
7.	Necht' je pro orientovaný graf $G=\langle H,U \rangle$ s ohodnocením hran w známo, že obsahuje cyklus se zápornou w -délkou dostupný z uzlu $s \in U$. Navrhnete efektivní algoritmus, který určí všechny uzly ležící na nějakém takovém cyklu.
8.	Modifikace Floyd-Warshallova algoritmu Proveďte úpravu algoritmu F-W tak, aby vedle matice vzdáleností D průběžně počítal i matici předchůdců na nejkratších cestách P . Jinou možností představuje výpočet matice P až na závěr algoritmu F-W z matice D . Navrhnete algoritmus složitosti $O(n^3)$ pro výpočet matice předchůdců P z matice D .
9.	Navrhnete algoritmus časové složitosti $O(H \cdot U)$ pro výpočet reflexivně-transitivního uzávěru orientovaného grafu $G=\langle H,U \rangle$. Návod: Upravte náležitě algoritmus F-W, resp. zvolte vhodný polookruh pro jeho obecnou verzi.
10.	Předpokládejme, že v Dijkstrově algoritmu není při implementaci prioritní fronty použita halda, ale jiná varianta implementace (např. pole $d[1..n]$ a fronta cyklicky v poli), která umožňuje vložení prvku i modifikaci priority v čase $O(1)$ a výběr nejmenšího prvku v čase $O(U)$. Zdůvodněte, že celková složitost algoritmu je potom $O(U \cdot U)$. Je možno symbol omikron ve výsledku nahradit symbolem Θ ?

11.	Předpokládejme, že v implementaci Dijkstrova algoritmu je graf reprezentován maticí incidence A grafu. Graf je navíc hustý, tj. $ H = \Theta(U ^2)$. Zdůvodněte, proč je pak jeho celková složitost $O(U ^3)$. Je možno symbol omikron ve výsledku nahradit symbolem Θ ?
12.	Maximální stupeň uzlu v grafu je 3. Zdůvodněte, proč je složitost Dijkstrova algoritmu aplikovaného na tento případ $O(U \cdot \log(U))$. Je možno symbol omikron ve výsledku nahradit symbolem Θ ?
13.	Graf je strom. Zdůvodněte, proč je složitost Dijkstrova algoritmu v takovém případě $O(U \cdot \log(U))$. Je možno symbol omikron ve výsledku nahradit symbolem Θ ?
14.	Otázky 10 až 13 přeformulujte a řešte pro Bellman-Fordův, Floyd-Warsahallův a Johnsonův algoritmus.