CVIČENÍ 9

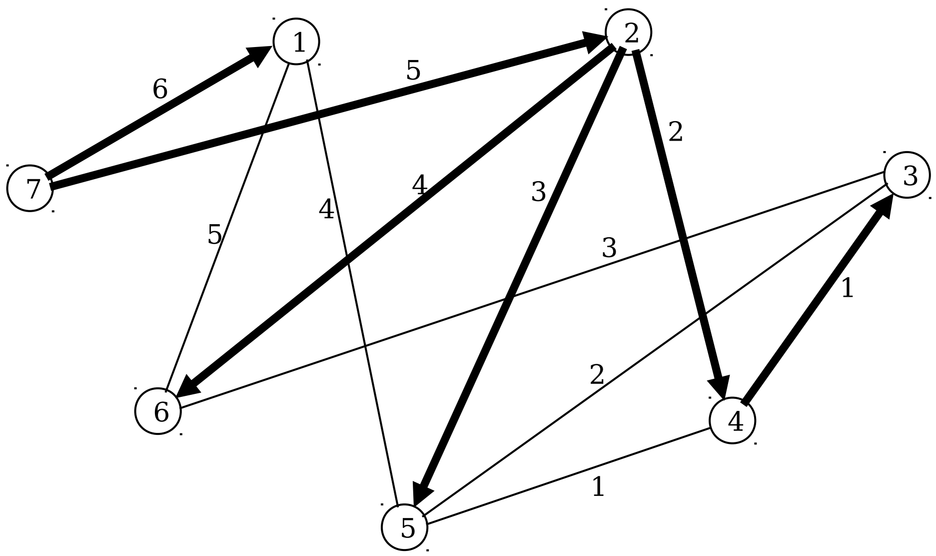
Téma: Nejkratší cesty
Cíle: Zvládnout techniku hledaní nejkratších cest v grafu, získat praktické zkušenosti se základními algoritmy (Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson). Pracovat se složitostmi těchto algoritmů a zvážit vliv úprav nebo omezení na speciální typy grafů na jejich složitost.

Nejkratší cesty	
1.	Vytvořte obyčejný neorientovaný graf G=⟨H,U⟩ : U= {1, 2, ... 7}, H = {[i,j] : 6 ≤ i+j ≤ 9}, tj. hrana mezi uzly i a j existuje právě tehdy, když 6 ≤ i+j ≤ 9. Nechť 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 ≤ i+j ≤ 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** Nechť je v obyčejném neorientovaném grafu G=⟨H,U⟩ zadáno ohodnocení w, které každé hraně [*u,v*] přiřazuje reálné číslo *w(u,v)* z intervalu ⟨0,1⟩. 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]* = -∞ všem uzlům u, pro které existuje cesta z uzlu *s* do uzlu *u* procházející cyklem se zápornou *w*-délkou.
7.	Nechť je pro orientovaný graf G=⟨H,U⟩ s ohodnocením hran *w* známo, že obsahuje cyklus se zápornou *w*-délkou dostupný z uzlu s∈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 současně počítal i matici

	předchůdců na nejkratších cestách P. Jinou možnost 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.	Navrhněte algoritmus časové složitosti O(|H|.|U|) pro výpočet reflexivně-tranzitivního uzávěru orientovaného grafu G=⟨H,U⟩. 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|·|U|). Je možno symbol omikron ve výsledku nahradit symbolem Theta?
11.	Předpokládejme, že v implementaci Dijkstrova algoritmu je graf reprezentován maticí incidence 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 Theta?
12.	Maximální stupeň uzlu v grafu je 3. Zdůvodněte, proč je složitost Dijkstrova algoritmu aplikovaného na tento případ O(|U|·log(|U|)). Je možno symbol omikron ve výsledku nahradit symbolem Theta?
13.	Graf je strom. Zdůvodněte, proč je složitost Dijkstrova algoritmu v takovém případě O(|U|·log(|U|)). Je možno symbol omikron ve výsledku nahradit symbolem Theta?
14.	Otázky 10 až 13 přeformulujte a řešte pro Bellman-Fordův, Floyd-Warsahallův a Johnsonův algoritmus.

CVIČENÍ 9 – možná řešení

Téma: Nejkratší cesty
Cíle: Zvládnout techniku hledaní 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=⟨H,U⟩ : U= {1, 2, ... 7}, H = {[i,j] : 6 ≤ i+j ≤ 9}, tj. hrana mezi uzly i a j existuje právě tehdy, když 6 ≤ i+j ≤ 9. Nechť 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 ≤ i+j ≤ 8 a pro ohodnocení hran platí w(i,j) = i+j. (Je možné doplnit i další varianty.)  **Druhý případ podobně ...**
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? **Vymyslete na místě ... Odpověď na otázku je, že by se ohodnocení nezměnilo.**
3.	**Bellman-Fordův algoritmus** Na jednoduchém příkladu zopakujte princip Bellmanova-Fordova algoritmu. **Vymyslete na místě a nezapomeňte na záporné hrany netvořící cyklus. Pak graf změňte, aby obsahoval záporný cyklus.**
4.	**Floyd-Warshallův algoritmus** Připomeňte strukturu Floyd-Warshallova algoritmu (promítněte/napište na tabuli jeho pseudokód). Diskutujte význam jednotlivých cyklů a dvojoperace uvnitř trojitého cyklu. **Vymyslete na místě – opět nejdříve bez záporného cyklu a pak**

změnit.

5. **Modifikace Dijkstrova algoritmu**

Nechť je v obyčejném neorientovaném grafu G=⟨H,U⟩ zadáno ohodnocení w, které každé hraně [*u*,*v*] přiřazuje reálné číslo *w(u,v)* z intervalu ⟨0,1⟩. 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.

Spolehlivost cesty je dána součinem ohodnocení jejích hran, vybíráme tedy cestu s maximálním ohodnocením.V Init-Paths se všechna d[u] inicializují na 0 a d[s] na 1. Pro uzavření se z fronty vybírá uzel s největší hodnotou d[u, v relax se testuje d[v] < d[u] * w(u,v) a v kladném případě se provede d[v] = d[u] * w(u,v) .

To odpovídá klasickému Dijkstrovu algoritmu pro ohodnocení w'(u,v) = – log(w(u,v))

6. **Modifikace Bellman-Fordova algoritmu**

Upravte Bellmanův-Fordův algoritmus tak, aby nastavil hodnoty *d[u]* = -∞ všem uzlům u, pro které existuje cesta z uzlu *s* do uzlu *u* procházející cyklem se zápornou *w*-délkou.

Místo řádek 5 až 8 vložit:

```
5   boolean result = true;
6   for ( i=1; i <= |U|-1; i++)
7     for ( Edge (u,v) in H(G) )
8       if ( d[v] > d[u] + w(u,v) ) {
9         d[v] = -∞; result = false; }
10  return result;
```

Hlavní cykl by šel zkrátit pomocí testu, zda v předchozím průchodu došlo k nějakému přiřazení.

7. Nechť je pro orientovaný graf G=⟨H,U⟩ s ohodnocením hran *w* známo, že obsahuje cyklus se zápornou *w*-délkou dostupný z uzlu s∊U. Navrhnete efektivní algoritmus, který určí všechny uzly ležící na nějakém takovém cyklu.

V algoritmu F-W provádím výpočet matice předchůdců a po výpočtu každého diagonálního prvků matice D se ptám, zda není záporný. Pokud ano, je obsažen v záporně ohodnoceném cyklu a pomocí odkazů na předchůdce získám příslušné uzly.

8. **Modifikace Floyd-Warshallova algoritmu**

Proveďte úpravu algoritmu F-W tak, aby současně počítal i matici předchůdců na nejkratších cestách P. Jinou možnost 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.

Tělo vnitřního cyklu změníme na tvar

$\texttt{if (} d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \texttt{) \{}$

$\quad d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)};\ p_{ij}^{(k)} = p_{kj}^{(k-1)};$

$\texttt{\}}$

Varianta s ex-post výpočtem matice předchůdců:

$\texttt{for (i=1; i<=n; i++)}$

$\quad\texttt{for (j=1; j<=n; j++)}$

$\qquad\texttt{for (k=1; k<=n; k++)}$

$\qquad\quad\texttt{if (} d_{ij} == d_{ik} + w_{kj} \texttt{) } p_{ij} = k;$

Vnitřní cyklus by šel ukončovat ihned po nalezení příslušné hodnoty

	k.
9.	Navrhněte algoritmus časové složitosti $O(\|U\|^3)$ pro výpočet reflexivně-tranzitivního uzávěru orientovaného grafu $G=\langle H,U\rangle$. Návod: Upravte náležitě algoritmus F-W, resp. zvolte vhodný polookruh pro jeho obecnou verzi. **Řešení je na slajdech z přednášky 8.**
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 Theta? **Uzel pro uzavření vyhledáme vždy výběrem (podle hodnoty v poli d) uzel z fronty s aktuálně minimální hodnotou d[u] (uzavřené uzly označujeme bitovou mapou). Modifikaci priority zajistí prosté přepsání hodnoty d[u], takže Relax má konstantní složitost. Na výběr uzlů k uzavření tedy potřebujeme celkem čas $O(\|U\|^2)$, na relaxaci všech hran $O(\|H\|)$. Hran je ale nejvýše $O(\|U\|^2)$, takže celková složitost je $O(\|U\|^2)$ a současně i $\Theta(\|U\|^2)$, neboť bez ohledu na počet hran potřebujeme čas $O(\|U\|^2)$ na výběr uzlů.**
11.	Předpokládejme, že v implementaci Dijkstrova algoritmu je graf reprezentován maticí incidence 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 Theta? **Relaxace sice proběhne |H|-krát, takže potřebuje čas $O(\|H\|)$ (nebo $O(\|H\|.\lg\|U\|)$ pro binární haldu), ale zjištění sousedů pro jeden uzel u bude vyžadovat průchod celého řádku a pak průchod $\delta(u)$ sloupců – celkem tedy v obou částech čas $O(\|U\|.\|H\|) = O(\|U\|^3)$ a současně i $\Theta(\|U\|^3)$.**
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 Theta? **V tomto grafu je tedy $\|H\| \le 1.5\ \|U\|$, neboli $\|H\| = O(\|U\|)$, což dává uvedenou celkovou složitost algoritmu i při použití binární haldy.**
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 Theta? **Ve stromu je $\|H\| = \|U\|-1$, neboli $\|H\| = O(\|U\|)$, což dává uvedenou celkovou složitost algoritmu i při použití binární haldy.**
14.	Otázky 10 až 13 přeformulujte a řešte pro Bellman-Fordův, Floyd-Warsahallův a Johnsonův algoritmus. **To jen, kdybyste se na závěr cvičení nudili ...**