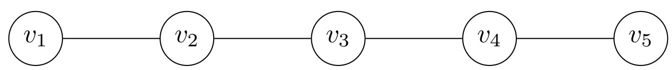
Cvičení z Algoritmů 3, 27. 10.

Vyřešte následující dva úkoly. Jejich řešení věnujte alespoň 90 minut (nebo méně, pokud se Vám je povede vyřešit dříve). Správná řešení zveřejním a okomentuji za týden. Není nutné mi nic posílat. V případě potřeby mě můžete kontaktovat mailem a můžeme si dohodnout konzultaci na Zoomu.

Úkol 1.
Pomocí algoritmu z přednášky nalezněte řešení úlohy batohu pro váhy položek $2, 5, 3, 4, 4, 1$ a kapacitu 8.

Úkol 2.
Nezávislá množina v neorientovaném grafu G je množina vrcholů tohoto grafu taková, že spolu žádné dva nesousedí.

V úkolu pracujeme s grafy, které jsou cestou. To znamená, že si je můžeme přestavit jako posloupnost vrcholů $v_1, v_2, v_3, \ldots, v_k$ a v grafu jsou jenom hrany mezi v_i a v_{i+1} pro $i = 1, 2, \ldots, k-1$. Příklad cesty s 5 vrcholy je na následujícím obrázku.



Dále každému vrcholu přiřadíme váhu, pro $i = 1, 2, \ldots, k$ je váhou vrcholu v_i kladné číslo w_i.

Úkol: navrhněte algoritmus, který v G nalezne nezávislou množinu tak, že součet vah vrcholů v ní obsažených je maximální (mezi všemi nezávislými množinami). Algoritmus musí pracovat v polynomickém čase a fungovat principem dynamického programování.

Řešení

Úkol 1.
Algoritmus vytvoří tabulku

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	2	2	2	2	2	2	2
2	0	0	2	2	2	5	5	7	7
3	0	0	2	3	3	5	5	7	8
4	0	0	2	3	4	5	6	7	8
5	0	0	2	3	4	5	6	7	8
6	0	1	2	3	4	5	6	7	8

Jedno z optimálních řešení je množina $\{2, 3\}$.

Úkol 2. Předpokládejme, že na vstupu máme graf G, který je cestou $v_1, v_2, \ldots, v_n$. Pomocí $I(i)$ označíme graf, která je tvořena prvními i uzly (tj. je to podgraf grafu G indukovaný množinou uzlů $\{v_1, v_2, \ldots, v_i\}$). Pomocí $OPT(i)$ označíme cenu optimálního řešení instance $I(i)$ (tedy sumu vah vrcholů v optimální nezávislé množině v grafu $I(i)$). Položíme si otázku: Patří vrchol v_i do optimálního řešení $I(i)$?

- pokud NE, pak je toto řešení rovno optimálnímu řešení instance $I(i-1)$,
- pokud ANO, pak je toto řešení rovno optimálnímu řešení instance $I(i-2)$, ke kterému přidáme vrchol v_i. (V nezávislé množině nemohou být sousední uzly, pokud je tam v_i, nemůže tam být v_{i-1}).

Instance lze uspořádat do orientovaného grafu (orientované cesty), s jediným zdrojem $I(0)$, počítáme v pořadí $I(0), I(1), I(2) \ldots, I(n)$. Přitom $OPT(0) = 0$. To jestli i patří do řešení nebo ne rozhodneme na základě toho, která z hodnot $OPT(i-1)$ a $OPT(i-2) + w_i$ je větší. Algoritmus má lineární složitost.