Záverečný test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/1.5	2/1.5	3/2	4/2	5/1.5	6/7	7/2	8/2	9/2	∑22.5

1. (1.5b) Pri hľadaní najkratších ohodnotených ciest v grafe je základnou operáciou relaxácia hrany. Nech $d[x]$ je aktuálny odhad dĺžky najkratšej ohodnotenej cesty (resp. sledu) zo štartovacieho vrcholu do vrcholu x a nech $p[x]$ je predchodca vrcholu x v nejakom slede s dĺžkou $d[x]$ zo štartovacieho vrcholu do vrcholu x. Ďalej nech $g[i][j]$ je ohodnotenie orientovanej hrany z vrcholu i do vrcholu j. Napíšte „zdrojový kód“ **operácie relaxácie** orientovanej hrany z vrcholu i do vrcholu j spolu s aktualizáciou obsahu poľa p:

2. (1.5b) Uvažujme kód z prednášky na prehľadávanie do šírky. Upravte ho tak, aby počítal vzdialenosť každého vrcholu od štartovacieho vrcholu.

```
public Map<Vertex, Boolean> bfs(Vertex start) {
	Map<Vertex, Boolean> navstiveny = g.createVertexMap(false);

	Queue<Vertex> rad = new LinkedList<>();
	navstiveny.put(start, true);
	rad.add(start);

	while (!rad.isEmpty()) {
		Vertex vrchol = rad.poll();
		for (Vertex sused : vrchol.getOutNeighbours()) {
			if (!navstiveny.get(sused)) {
				navstiveny.put(sused, true);
				rad.add(sused);
			}
		}
	}

	return navstiveny;
}
```

3. (2b) Na prednáške o dynamickom programovaní nás zaujímal problém výdaja cenných papierov za sumu peňazí m. Ukázalo sa, že riešenie založené na dynamickom programovaní spočíva vo vypĺňaní tabuľky, v ktorej máme minimálny počet cenných papierov, ktoré je nutné vydať na vyplatenie sumy m. Ak máme n rôznych hodnôt cenných papierov uložených v poli h, tak rekurzívny „vzorec“ vyjadrujúci optimálne riešenie na základe optimálnych riešení menších problémov (vyplatenia menších súm) je nasledovný:
   - R[0] = 0
   - R[s] = 1 + min {R[s – h[i]]|i = 0..n-1 také, že s – h[i] ≥ 0 a R[s – h[i]] ≠-1} resp. -1 ak je množina, z ktorej vyberáme minimum, prázdna.

   Zaujíma nás vyplatenie sumy 24 eur, pomocou cenných papierov s cenami 4, 7 a 9 eur. Tabuľka vyplnená pomocou rekurzívneho vzorca je uvedená nižšie. Pomocou tabuľky zistite, aké cenné papiere treba použiť na vyplatenie sumy 24 eur. Svoje riešenie zdôvodnite, t.j. zdôvodnite jednotlivé kroky výberu s odkazom na rekurzívny „vzorec“ a činnosť algoritmu.

   [0, -1, -1, -1, 1, -1, -1, 1, 2, 1, -1, 2, 3, 2, 2, 3, 2, 3, 2, 4, 3, 3, 3, 3, 4]

4. (2b) Uvažujem Dijkstrov algoritmus z prednášky. Stratili sme graf na ktorom sme spustili Dijkstrov algoritmus. Ale máme uchovaný obsah poľa vzdialeností vždy v momente, keď sme vybrali vrchol z množiny nevybavených hrán. Zostrojte graf, na ktorom by bol obsah poľa vzdialenosti počas behu Dijkstrovho algoritmu rovnaký.

A	B	C	D	E	F
∞	∞	0	∞	∞	∞
8	2	0	1	∞	∞
8	2	0	1	5	∞
7	2	0	1	5	3
6	2	0	1	4	3
5	2	0	1	4	3
5	2	0	1	4	3

5. (1.5b) Nakreslite súvislý neorientový graf s 5 hranami a 5 vrcholmi (označte vrcholy písmenami A-E), ktorého prechod prehľadávaním do šírky aj prechod prehľadávaním do hĺbky inicializovanými z vrcholu A prvý krát navštívia všetky vrcholy v rovnakom poradí. Predpokladáme, že susedné vrcholy „spracúvame“ v abecednom poradí.

Záverečný test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/1.5	2/1.5	3/2	4/2	5/1.5	6/8	7/2	8/2	9/2	∑22.5

6. (8b) Označte pravdivosť tvrdení (A – pravda, N – nepravda, +1b za správnu odpoveď, -1b za nesprávnu odpoveď, 0b za žiadnu odpoveď):

☐ (A) Máme ohodnotený neorientovaný graf s viac ako 10 vrcholmi ktorý má vždy práve 2 hrany s rovnakou váhou. Potom pre každý takýto graf existujú aspoň 2 rôzne najlacnejšie kostry.

☐ (B) Majme orientované grafy G1 a G2, na oboch existuje topologické usporiadanie. Graf G3 vznikne zjednotením grafov G1 a G2 a pridaním jednej orientovanej hrany z ľubovoľného vrcholu z G1 do ľubovoľného vrcholu z G2. Potom existuje topologické usporiadanie na orientovanom grafe G3.

☐ (C) Topologické usporiadanie na grafe skončilo s neprázdnym grafom. Potom v tomto neprázdnom grafe existuje vrchol z ktorého existuje cesta do každého vrchola ktorý ostal v grafe.

☐ (D) Ak v prehľadávaní do hĺbky ohodnoteného grafu budeme navštevovať susedné vrcholy aktuálneho vrcholu v poradí podľa ceny hrán (t.j. v každom kroku vyberieme nenavštívený susedný vrchol taký, že cena hrany, ktorá spája aktuálny a susedný vrchol, je najmenšia možná), potom vytvorená DFS kostra je zároveň najlacnejšou kostrou.

☐ (E) Memoizácia (pamätanie si a znovupoužitie výsledkov podvýpočtov) umožňuje znížiť časovú zložitosť každého rekurzívneho výpočtu.

☐ (F) Z terminológie neohodnotených grafov vieme, že excentricita vrcholu grafu je vzdialenosť od neho k najvzdialenejšiemu vrcholu a centrum grafu je množina vrcholov s minimálnou excentricitou. Potom pre všetky vrcholy X, Y z centra grafu platí $(X, Y) \in E(G)$.

☐ (G) Pre každý súvislý ohodnotený graf, v ktorom majú všetky hrany navzájom rôzne ceny, platí, že má vždy práve jednu najlacnejšiu (minimálnu) kostru.

☐ (H) Uvažujme prehľadávanie súvislého grafu algoritmom do šírky resp. do hĺbky. Hrany, ktorými sme nejaký vrchol navštívili po prvý raz tvoria kostru. Kostra vytvorená prehľadávaním do šírky má vždy rovnaký počet hrán ako kostra vytvorená prehľadávaním do hĺbky.

7. (2b) Uvažujme algoritmy uvedené dole predpokladajme, že graf má *n* vrcholov a *m* hrán, pomocou O notácie alebo Θ notácie vyjadrite zložitosť algoritmov (+0.5b za správnu odpoveď -0.5b za nesprávnu, 0b za žiadnu odpoveď).

   Bellman-Ford algoritmus

   Dijkstrov algoritmus

   Floyd-Warshall algoritmus

   Primov (Jarníkov) algortimus

8. Implementáciou `HashSet`-u sme sa zaoberali na prednáške. Každému reťazcu bol priradený jeho index hash-ovacou funkciou ako súčet unicode hodnôt znakov modulo veľkosť poľa. Predpokladajme, že zmeníme hash-ovaciu funkciu, ktorá index vypočíta ako dĺžku reťazca modulo veľkosť poľa.

   (1b) Uvažujeme uvedenú implementáciu s poľom veľkosti 4. Napíšte 3 slová ktoré budú mať vzájomné kolízie a dĺžky slov budú rozdielne.

   _____

   (1b) Koľko slov treba vložiť do uvedenej implementácie `HashSet`-u, ak má interne pole veľkosti 16 a na začiatku je prázdny, aby bol load factor práve 2?

   _____

9. (2b) Uvažujme neorientovaný graf reprezentujúci mesto, kde vrcholy sú križovatky a cesty sú hrany medzi nimi. Mesto sa rozhodlo, že chce umiestniť na vybrané križovatky kamery tak, aby každá cesta bola monitorovaná (buď má kameru na jednej križovatke alebo oboch). Všetky cesty musia byť monitorovane a zároveň chce mesto umiestniť čo najmenej kamier. Preto Franklin vymyslel nasledujúci algoritmus:

   1. Usporiada si vrcholy podľa počtu ich susedov bez kamery.
   2. Na vrchol s najväčším počtom susedov bez kamery umiestni kameru.
   3. Ak je ešte cesta ktorá nie je monitorovaná vráti sa na bod 1.

   Zdôvodnite, prečo tento výber kamier dokáže monitorovať celé mesto pomocou najmenšieho počtu kamier, alebo nájdite kontrapríklad.