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	2/2.5	3/2	4/2	5/4.5	6/2	7/2.5	8/2	9/2	10/2	∑23

1. (1.5b) Nech **g** je matica susednosti súvislého neorientovaného neohodnoteného grafu bez slučiek a v poli **d** chceme pre každý vrchol grafu určiť jeho vzdialenosť (meranú počtom hrán) od iniciálneho vrcholu prehľadávania do šírky (BFS) určeného vrcholom s číslom **start**.

```
int[] d = new int[g.length];
for (int i = 0; i < d.length; i++)
      d[i] = -1;
Queue<Integer> rad = new LinkedList<Integer>();
d[start] = 0;
rad.offer(start);
while (!rad.isEmpty()) {
      int i = rad.poll();
      for (int j = 0; j < d.length; j++)

            if (__________________________________) {
                  d[j] = d[i] + 1;
                  rad.offer(j);
            }
}
```

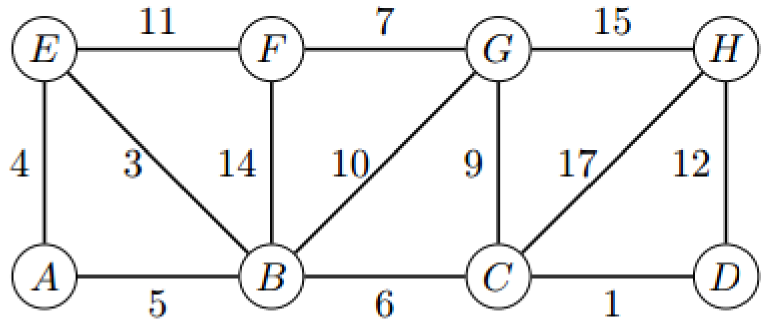
Na vyznačené miesto doplňte podmienku, ktoré zabezpečí, že „navštívime" všetky doposiaľ „nenavštívené" susedné vrcholy vrcholu s číslom i.

2. (2.5b) Nech S je množina intervalov $S = \{I_1, I_2, I_3, \dots, I_n\}$, pričom $I_k = \langle a_k, b_k \rangle$. Chceme vybrať najväčšiu (počtom prvkov - intervalov) takú podmnožinu Q množiny S, $Q \subseteq S$, že pre ľubovoľné 2 prvky $I_i, I_k \in Q$ platí $I_i \cap I_k = \emptyset$. Teda žiadne 2 intervaly z Q nemajú spoločný prienik. Uvažujme nasledovný greedy algoritmus:

$Q \leftarrow \emptyset$;
kým (S nie je prázdna) {
  vyber taký interval $I \in S$, ktorý má najmenší možný začiatok spomedzi intervalov v S;
  presuň I z množiny S do Q, t.j. $Q \leftarrow Q \cup \{I\}$ a $S \leftarrow S \setminus \{I\}$;
  odstráň z množiny S všetky intervaly, ktoré majú neprázdny prienik s I;
}

Dokážte, že algoritmus vráti korektný výsledok alebo nájdite kontrapríklad.

Graf k úlohám 3-5.



3. (2b) Určte množinu hrán, ktoré vyberie Primov algoritmus na nájdenie najlacnejšej kostry so štartovacím vrcholom A. **Hrany** zapíšte v poradí, v akom ich tento algoritmus pridá do konštruovanej kostry.

4. (2b) Ak prehľadávanie do hĺbky je inicializované vo vrchole A a susedné vrcholy „spracúvavame“ vždy v abecednom poradí, zapíšte postupnosť vrcholov grafu, v akej budú navštívené:

5. Uvažujme beh Dijkstrovho algoritmu pričom hľadáme najlacnejšie cesty z vrcholu A.
(2b) V akom poradí sa jednotlivé vrcholy stanú „vybavenými“?

(1.5b) Dijkstrov algoritmus vieme upraviť tak, aby sme si pre každý vybavený vrchol zapamätali jeho predchodcu (vrchol, z ktorého sme „prišli“ do daného vrcholu). Pre každý jeden vrchol v nasledujúcej tabuľke vyplňte jeho predchodcu na najkratšej ceste z vrcholu A.

(1b) V tabuľke nižšie pre každý vrchol určte hodnotu horného ohraničenia ceny najkratšej cesty v okamihu, keď sa tento vrchol stane „vybaveným“.

	A	B	C	D	E	F	G	H
predch.								
d								

6. (2b) Kľúčovým prvkom Knuth-Morris-Prattovho algoritmu je skip matica, ktorá pre aktuálnu pozíciu vo vzorke a písmeno na vstupe rozhoduje, o koľko pozícií sa má posunúť začiatok vzorky vo vstupe (narozdiel od naivného algoritmu, kde sa pri nezhode posúva začiatok vzorky vždy o 1 pozíciu doprava). Vyplňte skip maticu pre zadanú vzorku a množinu písmen vzorky. Stĺpce odpovedajú aktuálnej pozícii vo vzorke a riadky písmenu, ktoré je na vstupe.

	X	Y	Y	X	X
X					
Y			0		
iné písmeno	1		3		

7. (2.5b) Označte pravdivosť tvrdení (A – pravda, N – nepravda, +0.5b za správnu odpoveď, -0.5b za nesprávnu odpoveď, 0b za žiadnu odpoveď):

☐ 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.

☐ (A) Kruskalov algoritmus na nájdenie najlacnejšej kostry funguje korektne aj v prípade, že ohodnotenia hrán sú záporné.

☐ (B) Ak graf obsahuje nejaké 2 hrany s rovnakým ohodnotením, potom má vždy viac než len jednu najlacnejšiu kostru.

☐ (C) Uvažujme Dijkstrov algoritmus na nájdenie najlacnejších ciest. V okamihu, keď pri relaxácií hrán vychádzajúcich z vybraného vrcholu nedôjde k zmenám horných ohraničení jeho susedov, algoritmus môžeme ukončiť, pričom platí, že všetky zatiaľ nevybrané vrcholy už sú vybavené, t.j. ich horné ohraničenia sa rovnajú cenám najlacnejších doň vedúcich ciest.

☐ (D) Karp-Rabinov algoritmus je založený na hashovaní. Využíva sa pritom, že výskyt vzorky v reťazci je identifikovaný len na základe rovnosti hash-ov vzorky a prislúchajúceho podreťazca bez nutnosti porovnávať tento podreťazec a vzorku po písmenách.

8. (2b) Zdôvodnite svoju odpoveď ku ktorémukoľvek tvrdeniu A-D z predošlej úlohy. Odôvodňované tvrdenie označte zakrúžkovaním písmena.

9. (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. <u>Svoje riešenie zdôvodnite</u>, 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]

10. (2b) Rozhodnite o pravdivosti tvrdenia a svoje rozhodnutie vyargumentujte:

V každom ohodnotenom n-vrcholovom grafe:

- s nezápornými cenami,
- s $n > 10$,
- v ktorom sa každá cena hrany objavuje iba konštantný počet raz (napr. 1, 2, 3, 4, …, ale nie n, či $n/2$ – t.j. napríklad graf, v ktorom majú všetky hrany rovnakú cenu, nespĺňa túto podmienku)

existuje medzi každými 2 vrcholmi <u>nanajvýš</u> $4.n^3$ rôznych najlacnejších ciest.