Záverečný test

teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/2	2/3	3/2	4/4	5/2	6/6	7/1	8/2	∑11/22

1. (2b) Nakreslite orientovaný graf so 6 vrcholmi označenými A, B, C, D, E, F a 12 hranami, taký, že topSort skončí s neprázdnym grafom ktorý ma vrcholy D, E, F. Napíšte v akom poradí budú jednotlivé vrcholy odstránené na vami nakreslenom grafy.

2. (3b) Popis problému: „Hodnotitelia Viktor a Ďuri budú hodnotiť teoretický finalterm, kde sú úlohy za nasledujúce bodové zisky: 2, 3, 2, 4, 2, 6, 1, 2 (vždy celé čísla). Každý z nich hodnotí vždy tú istú úlohu a úloha má jedného hodnotiteľa. Chcú si úlohy rozdeliť tak, aby každý hodnotil úlohy za rovnaký počet bodov.“
Navrhnite/popíšte algoritmus, ktorý by zistil, či takéto rozdelenie existuje. Váš algoritmus má fungovať aj pre iné vstupy ako je vzorový. Ak používate časť alebo celý alebo modifikujete algoritmus z prednášky alebo cvičenia môžete z neho vychádzať ale uveďte ktorý algoritmus alebo úlohu používate. Napíšte ku ktorej téme patrí vami navrhnutý algoritmus (napr. orientované grafy, greedy, dynamické programovanie). Body závisia aj od triedy zložitosti vami navrhnutého algoritmu.

3. (2b) Uvažujme Floyd-Warshallov algoritmus na hľadanie najlacnejších ciest v grafe. Dole je uvedená časť kódu. Popíšte čo vo všeobecnosti vypočíta po jednej iterácii prvého (zvýrazneného) cyklu, teda pre k=0.

```
for (int k = 0; k < vrcholy.length; k++)
     for (int i = 0; i < vrcholy.length; i++)
          for (int j = 0; j < vrcholy.length; j++)
               if (d[i][k] + d[k][j] < d[i][j])
                    d[i][j] = d[i][k] + d[k][j];
```

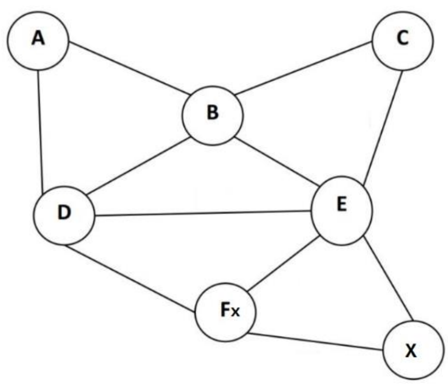
4. (4b) Uvažujme BFS a DFS, ktoré vyberajú vrcholy v abecednom (lexikografickom) poradí. Pre graf na obrázku napíšte:

Poradie navštívenia vrcholov pre DFS so štartovacím vrcholom E:

Vrcholy tvoriace centrum grafu:

Priemer grafu:

Poradie v ktorom budú pridávané **hrany** do BFS kostry so štartovacím vrcholom D:



5. (2b) Dopíšte kód metódy, ktorá pre pole s hodnotami cenných papierov 1000, 250, 50 vráti pole, kde budú počty príslušných cenných papierov, potrebných na vyplatenie sumy (zadanej parametrom). Súčet počtu cenných papierov má byť minimálny Pozn.: Ak použijete greedy algoritmus, zdôvodnite jeho korektnosť. Ak na danú úlohu neexistuje greedy algoritmus, použite iný, časovo čo najefektívnejší, prístup (dynamické programovanie, backtracking) a označte časovú zložitosť riešenia.

```java
public static int[] poctyCP(int suma) {
      int[] cp = {1000,250,50};
      int[] poctyCP = {0,0,0};

      return poctyCP;
}
```

6. (6b) Označte pravdivosť tvrdení (A - pravda, N - nepravda, +1,5b za správnu odpoveď so zdôvodnením +0.5b za správnu odpoveď bez zdôvodnenia, -0.5b za nesprávnu odpoveď bez zdôvodnenia, 0b za žiadnu odpoveď alebo nesprávnu odpoveď s „rozumným“ zdôvodnením):

☐ **(A)** Ak centrum (súvislého) grafu je tvorené viac ako jedným vrcholom, platí, že ľubovoľné dva vrcholy z centra grafu sú navzájom susedné.

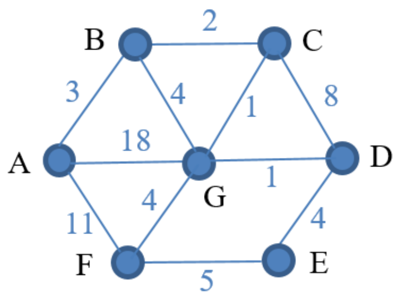
☐ **(B)** Majme ohodnotený neorientovaný graf G s *n* vrcholmi. Graf H vznikne z grafu G tak, že odstránime všetky ohodnotenia hrán, teda je neohodnotený. Označme *d* priemer grafu H. Bellman-Fordov algoritmus v každej fáze relaxuje všetky hrany grafu a fáz je *n*. Je pravda, že stačí aby Bellman-Fordov algoritmus aplikovaný na grafe G vykonal *d* fáz?

☐ **(C)** Kruskalov algoritmus na súvislom grafe s $n \geq 3$ vrcholmi začne s *n* komponentmi. Je pravda, že po vybavení prvých dvoch hrán bude počet komponentov vždy *n-2*?

☐ **(D)** 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.

7. (1b) Pozrite si bodovanie v úlohe 6. Položme si otázku „Je možné získať akýchkoľvek x bodov v rozsahu od +6 bodov do -2 bodov, kde x je násobok 0,5b.“ Je možné vo všeobecnosti dať odpoveď na túto otázku v čase lepšom ako exponenciálnom, teda použitím backtracku? Odpoveď zdôvodnite/argumentujte.

8. (2b) Uvažujme Dijkstrov algoritmus, ktorý už označil prvé dva vrcholy A a B za vybavené. Hodnoty poľa po vybavení týchto vrcholov, sú uvedené pri grafe. Odsimulujte tento algoritmus kým nevybavíte ďalšie 2 vrcholy. Napíšte aké budú hodnoty vzdialenosti po vybavení ďalších dvoch vrcholov. Napíšte ktorý vrchol bol štartovací.



$d_s[]$	A	B	C	D	E	F	G
	0	3	5	∞	∞	11	7

Štartovací vrchol:

ds[]	A	B	C	D	E	F	G