Záverečný test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/2b	2a/1b	2b/2b	2c/0.5	∑5.5b

1. (1.5) Nakreslite graf, na ktorom boli počas behu BFS (prehľadávanie do šírky) a DFS (prehľadávanie do hĺbky) prvýkrát navštívene vrcholy v danom poradí, pričom oba algoritmy navštevujú susedov v abecednom poradí.

   BFS: A, C, D, E, F, B

   DFS: A, C, E, B, D, F

   (0.5b) Pre vami nakreslený graf určte:
   Maximálnu veľkosť radu (počet vrcholov v rade)
   počas výpočtu BFS:

   Maximálnu veľkosť zásobníka (počet vrcholov v zásobníku)
   počas výpočtu DFS:

2. Uvažujme nasledujúci problém. Máme pole celých čísel *p*. Hľadáme súčet vybranej podpostupnosti čísel, pričom chceme aby:
   -súčet prvkov podpostupnosti bol maximálny,
   -dvojica prvkov vedľa seba nemôže byť vybraná.

   Príklad: Pre pole uvedené v bode a) je vybraná podpostupnosť napr. indexy 1, 3, 9, resp. hodnoty 3, -3, 7, ich súčet je 7. Táto postupnosť nie je maximálna.

   Uvažujme riešenie pomocou dynamického programovania. V poli *dp* na index *i* uložíme maximálnu hodnotu vybranej podpostupnosti, ktorá obsahuje prvok *i* z poľa *p*.

   a) (1b) Pre uvedené pole *p* vyplňte hodnotu poľa *dp*.

i:	0	1	2	3	4	5	6	7	8	9	10	11	12
p:	-2	3	-1	-3	9	6	1	5	-2	7	7	-3	-5

i:	0	1	2	3	4	5	6	7	8	9	10	11	12
dp:													

   b) (2b) Napíšte tzv. rekurzívny „vzorec“ vyjadrujúci hodnotu *dp[i]*:

   c) (0.5b) Popíšte, ako nájsť v poli *dp* súčet vybranej podpostupnosti čísel, ktorej súčet je maximálny:

Meno a priezvisko:	Skupina PAZ:	

3a/1b	3b/1b	3c/0.5	3d/2b	4/1b	∑5.5b

3. Zoberme beh Bellman-Fordovho algoritmu, ktorý na ohodnotenom neorientovanom grafe **po dvoch relaxáciách všetkých hrán** je v stave uvedenom tabuľkou vzdialenosti *d*. Vieme, že každá hrana e, ktorá je v grafe G má celočíselné ohodnotenie $3 \leq c(e) \leq 9$ a c(A,B)=3.

	A	B	C	D	E	F	G	H
d:	0	3	7	7	5	∞	∞	4

Na každú otázku odpovedajte nezávisle od predchádzajúcich otázok.

a) (1b) Aká môže byť minimálna a maximálna vzdialenosť medzi vrcholmi B a C?

min: max:

b) (1b) Ak navyše vieme, že vrchol A určite susedí aj s vrcholom D, s ktorými ďalšími vrcholmi musí určite susediť?

c) (0.5b) Ak predpokladáme, že graf G je **súvislý**, aká môže byť najväčšia vzdialenosť vrcholu F po dokončení Bellman-Fordovho algoritmu so štartovacím vrcholom A?

d) (0.5+1.5b) Nakreslite súvislý graf, ktorý spĺňa podmienky zadania (tabuľka d, $3 \leq c(e) \leq 9$ a c(A,B)=3). Odsimulujte na ňom Primov algoritmus.

	A	B	C	D	E	F	G	H	Q množina nevybavených vrcholov
d:	0	∞	∞	∞	∞	∞	∞	∞	ABCDEFGH

4. (1b) Napíšte kód, ktorý pre neorientovaný neohodnotený graf G reprezentovaný maticou susednosti *gm* vráti maximálny stupeň vrcholu v grafe.

```
public int maxVertexDegree(boolean[][] gm) {
```

Meno a priezvisko:	Skupina PAZ:	

5/1b	6/6b	7/2b	8/2b	∑11b

5. (1b) Uvažujme Floyd-Warshallow algoritmus na hľadanie najlacnejších ciest v grafe. Pre ktoré priradenia premenných i, j, k symbolom vo for-cykloch bude algoritmus fungovať korektne, t.j. korektne sa naplní obsah matice d?

(a) len Δ=k,▼=i,☺=j a Δ=i,▼=j,☺=k
(b) len Δ=k,▼=i,☺=j
(c) Δ=k,▼=i,☺=j Δ=k,▼=j,☺=i Δ=i,▼=j,☺=k
(d) Δ=k,▼=i,☺=j Δ=i,▼=j,☺=k Δ=j,▼=k,☺=i
(e) Δ=k,▼=i,☺=j a Δ=k,▼=j,☺=i

```
for (int Δ = 0; Δ < n; Δ++)
  for (int ▼ = 0; ▼ < n; ▼++)
    for (int ☺= 0; ☺< n; ☺++)
      if (d[i, k] + d[k, j] < d[i, j])
        d[i, j] = d[i, k] + d[k, j];
```

6. (6b) Označte pravdivosť tvrdení (A - áno/pravda, N - nie/nepravda, +1b za správnu odpoveď, -0.5b za nesprávnu odpoveď, 0b za žiadnu odpoveď):

☐ (A) Môže mať graf viac topologických usporiadaní ako má vrcholov?

☐ (B) V ohodnotenom n-vrcholovom grafe s nezápornými cenami existuje medzi každými 2 vrcholmi nanajvýš $O(n^3)$ rôznych najlacnejších ciest. Tento fakt nepriamo využíva aj Floyd-Warshallov algoritmus.

☐ (C) Hľadáme všetky vrcholy grafu, do ktorých najlacnejšia cesta zo zadaného vrcholu s má cenu nanajvýš 50. Ohodnotenia hrán sú nezáporné. Aplikujeme Dijkstrov algoritmus. V okamihu, keď počas behu algoritmu prehlásime za vybavený taký vrchol, ktorého cena (vzdialenosť) od vrcholu s je väčšia ako 50, potom algoritmus môžeme ihneď ukončiť. Pritom platí, že všetky hľadané vrcholy budú medzi vrcholmi, ktoré sme do daného okamihu prehlásili za vybavené.

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

☐ (E) Majú Bellman-Fordov algoritmus a Dijsktrov algoritmus rovnakú časovú zložitosť vzhľadom k počtu vrcholov *n* v O notácii?

☐ (F) V Kruskalovom algoritme hrany vybrané do minimálnej kostry tvoria vždy (počas celého vykonávania algoritmu) súvislý podgraf.

7. (2b) Označte 2 tvrdenia z úlohy 6 a napíšte k nim zdôvodnenie.

8. Milan sa rozhodol investovať do striebra. Začína s 37€. Každý deň môže kúpiť striebro alebo ho predať, pričom vie nakupovať iba celé gramy. Odhad vývoja cien je nasledovný:

Dátum	11.5.	12.5.	13.5	14.5.	15.5.	16.5.	17.5.	18.5.	19.5.
€/g	35	33	31	36	34	33	39	41	38

(0.5b) Nanajvýš koľko € môže mať 19.5.?

(1.5b) Popíšte optimálny greedy algoritmus na nákup a predaj striebra, aby bol zisk v € na konci obdobia maximálny. Hint: Kedy nakupovať a kedy predávať?