Záverečný test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/3.5	2/1.5	3/4	4/3	5/2.5	6/2.5	7/1.5	8/1	9/1	10/2	∑22.5

1. (2b+1.5b) Majme ohodnotený neorientovaný graf, ktorého matica susednosti je nasledujúca (symbol ∞ znamená neprítomnosť hrany):

	A	B	C	D	E	F	G
A	∞	7	∞	5	∞	∞	∞
B	7	∞	8	9	7	∞	∞
C	∞	8	∞	∞	5	∞	∞
D	5	9	∞	∞	15	6	∞
E	∞	7	5	15	∞	8	9
F	∞	∞	∞	6	8	∞	11
G	∞	∞	∞	∞	9	11	∞

Napíšte, v akom poradí sa jednotlivé vrcholy stanú „vybavenými“, ak na daný graf aplikujeme Dijkstrov algoritmus so štartovacím vrcholom C:

____

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 aplikovaním Dijkstrovho algoritmu so štartovacím vrcholom C:

	A	B	C	D	E	F	G
C			X				

2. (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:

3. (4b) Označte pravdivosť tvrdení (A – pravda, N – nepravda, +0.5b za správnu odpoveď, -0.5b za nesprávnu odpoveď, 0b za žiadnu odpoveď):

☐ Dijkstrov algoritmus vyberá z množiny potenciálne nevybavených vrcholov taký vrchol, ktorého horné ohraničenie ceny doň vedúcej najlacnejšej cesty je minimálne. Vybraný vrchol je prehlásený za vybavený. Ak by sme pri každom prehlásení vrcholu za vybavený vypísali jeho aktuálne horné ohraničenie, dostali by sme neklesajúcu postupnosť.

☐ Nech $c(X,Y) \geq 0$ je ohodnotenie orientovanej hrany (X,Y). Ak v grafe pre ľubovoľnú trojicu vrcholov $A,B,C \in V(G)$ takých, že $(A,B),(B,C),(A,C) \in E(G)$, platí $c(A,B) + c(B,C) > c(A,C)$, potom najkratšiu cestu medzi dvoma vrcholmi možno nájsť prehľadávaním do šírky.

☐ Algoritmus Knuth-Morris-Pratt má rovnakú asymptotickú časovú zložitosť ako naivný algoritmus, má však menšiu pamäťovú zložitosť.

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

☐ Každý program využívajúci rekurziu má vždy aspoň exponenciálnu zložitosť.

☐ Greedy algoritmy sú založené na myšlienke uchovania si riešení podproblémov, ktoré sme už riešili, a ich použití pri ich opakovanom výskyte.

☐ 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.

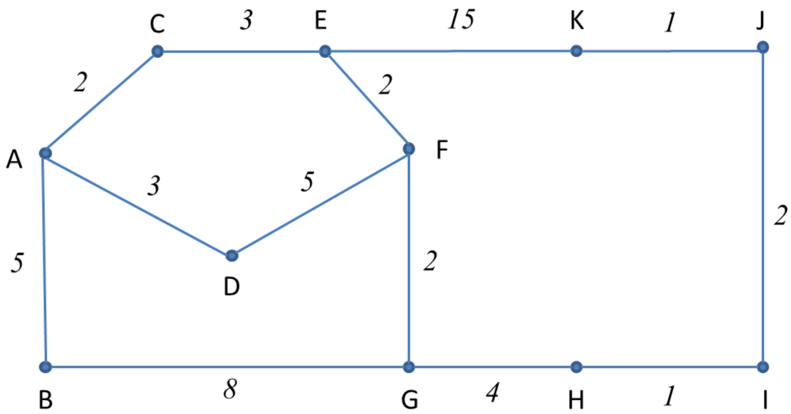
☐ Ak v Kruskalovom algoritme hrany usporiadame od najdrahšej po najlacnejšiu, výsledkom algoritmu bude najdrahšia kostra grafu.

4. (3b) Pole P dĺžky n obsahuje permutáciu čísel $0, \dots, n-1$ (t.j. každé číslo z tohto rozsahu sa v poli P nachádza práve raz). Navrhnite a zapíšte kód, ktorý v čase $O(n)$ vytvorí také pole R dĺžky n, že platí: $R[x] < R[y]$ práve vtedy, ak sa hodnota x v poli P nachádza na menšom indexe ako hodnota y ($x, y \in \{0, \dots, n-1\}$).

5. (2.5b) Pred výťahom s maximálnou nosnosťou M, stojí n osôb s váhami $m_1, m_2, \dots, m_n$ ($\forall i, m_i \leq M$). Cieľom je všetky osoby prepraviť výťahom. Uvažujme takýto greedy algoritmus:
   1. Vyberieme takú najťažšiu osobu čakajúcu pred výťahom, ktorá po nastúpení do výťahu nespôsobí prekročenie nosnosti. Ak takej osoby niet, výťah odíde a ide sa na krok 2, inak sa ide krok 1.
   2. Ak pred výťahom ešte niekto stojí, privolá sa výťah a ide sa na krok 1.

   Vedie tento greedy algoritmus k riešeniu s minimálnym počtom jázd výťahu? Vyargumentujte korektnosť algoritmu alebo nájdite kontrapríklad, že tento algoritmus nevedie k optimálnemu riešeniu.

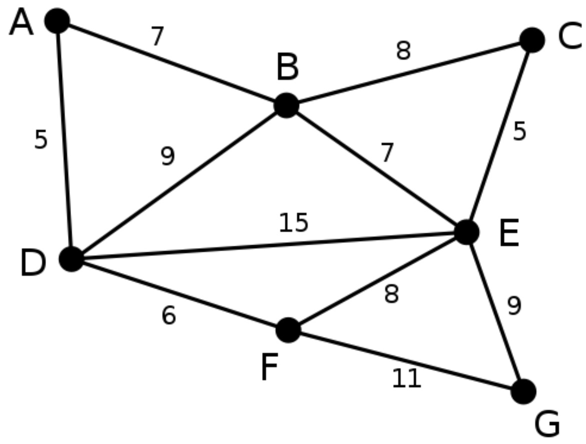
6. (2.5b) Uvažujme problém nájdenia najlacnejšej cesty z vrcholu A do všetkých ostatných vrcholov grafu. Nájdite také usporiadanie hrán grafu, že zrelaxovanie hrán grafu v tomto poradí bude mať za následok, že všetky vrcholy grafu sa stanú „vybavené“. Keďže graf je neorientovaný, pod relaxovaním neorientovanej hrany chápeme relaxovanie dvoch protichodne orientovaných hrán (s rovnakou cenou), ktoré táto hrana reprezentuje.



7. (1.5b) Nakreslite orientovaný graf, ktorý má <u>práve</u> 5 rôznych topologických usporiadaní vrcholov.

8. (1b) Nech pole [20, 12, 10, 8, 7, 5, 4, 1] reprezentuje binárny strom, ktorý je haldou. Nech k je výška daného stromu. Aký minimálny počet uzlov je potrebné pridať do stromu tak, aby mal výšku k + 1 (resp. aby sa jeho výška zväčšila o 1) a zároveň, aby ešte stále ostal haldou?

Graf k úlohám 9 a 10:



9. (1b) Ak prehľadávanie do šírky je inicializované vo vrchole A, zapíšte postupnosť vrcholov grafu, v akej budú navštívené:

10. (2b) Zapíšte postupnosť hrán grafu, v akej budú pridané jednotlivé hrany grafu do minimálnej kostry pri aplikovaní Kruskalovho algoritmu: