Záverečný test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/1	2/2	3/3	4/2	5/2.5	6/2	7/1.5	8/4	9/1.5	10/1	11/2	Σ22.5

1. (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];
```

2. (2b) Uvažujme Karp-Rabinov algoritmus, ktorý pri vyhľadávaní všetkých výskytov vzorky vo vstupe využíva haše vzorky a príslušných podreťazcov vstupu. Nájdite takú vzorku a taký vstup, aby Karp-Rabinov algoritmus mal najväčšiu možnú časovú zložitosť nezávisle od použitej hašovacej funkcie (=postupu, ako sa vyráta haš pre reťazec).

   Vstup (15 znakov):

   Vzorka (5 znakov):

	C			

3. (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 ($x, y \in \{0, \dots, n-1\}$).

4. (2b) Metóda `pisomka` dostane ako parameter *g* maticu susednosti orientovaného grafu bez slučiek.

```
public static int pisomka(boolean[][] g) {
  int r = 0;
  for (int i = 0; i < g.length; i++) {
    for (int j = i+1; j < g.length; j++)
      if (g[i][j] && g[j][i])
        r++;
  }

  return r;
}
```

Nakreslite taký orientovaný graf so 6 vrcholmi, že metóda `pisomka` vráti číslo 2 a zároveň každý vrchol grafu je koncom alebo začiatkom aspoň jednej orientovanej hrany.

5. (2.5b) Algoritmus na topologické usporiadanie vrcholov orientovaného grafu v každom kroku odstráni z grafu vrchol, do ktorého nevchádza žiadna orientovaná hrana. Ak takého vrcholu niet a graf obsahuje aspoň jeden vrchol, potom graf obsahuje aspoň jeden cyklus. Nakreslite taký 6-vrcholový orientovaný graf, pre ktorý platí:
   - do každého vrcholu vchádza aspoň jedna orientovaná hrana a
   - graf obsahuje aspoň jednu orientovanú hranu, ktorá nepatrí do žiadneho orientovaného cyklu

6. (2b) Kniha cvikov obsahuje zoznam rôznych fitness cvikov. O každom cviku vieme, koľko kalórii sa pri ňom spáli a tiež aké je trvanie cviku v minútach. Fitness trenér chce zostaviť M-minútový tréning (tréningový plán), pričom každý cvik chce počas tréningu realizovať nanajvýš raz. Zároveň chce, aby sa počas tohto tréningu spálilo čo najviac kalórii. Rozhodol sa použiť takýto algoritmus:
   - `Vyber doposiaľ nevybraný cvik, ktorý spáli najviac kalórii a zároveň sa ešte celý zmestí do zostávajúceho času tréningu. Ak takého cviku niet, tréningový plán je zostavený. V opačnom prípade pokračuj výberom ďalšieho cviku podľa rovnakého algoritmu.`

Vedie tento algoritmus k optimálnemu riešeniu? Zdôvodnite alebo nájdite kontrapríklad.

7. (1.5b) Pri hľadaní najlacnejších ohodnotených ciest v grafe je základnou operáciou relaxácia hrany. Nech $d[x]$ je aktuálny odhad ceny najlacnejšej ohodnotenej cesty (resp. sledu) zo štartovacieho vrcholu do vrcholu x a nech $p[x]$ je predchodca vrcholu x v nejakom slede s cenou $d[x]$ 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:

8. (4b) Označte pravdivosť tvrdení (A – pravda, N – nepravda, +0.5b za správnu odpoveď, -0.5b za nesprávnu odpoveď, 0b za žiadnu odpoveď):

☐ Dijkstrov algoritmus (podobne ako Bellman-Fordov algoritmus) funguje korektne aj v prípade, že ohodnotenia hrán sú záporné.

☐ 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.

☐ 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.

☐ 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 najlacnejšiu cestu medzi dvoma vrcholmi možno nájsť prehľadávaním do šírky.

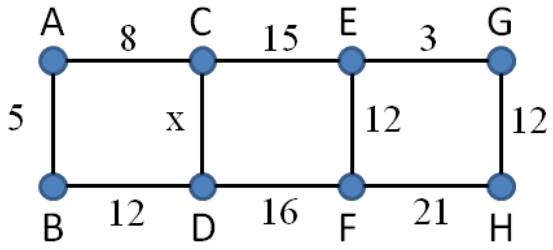
☐ Základná myšlienka Bellman-Fordovho algoritmu je v orientovanom grafe s n vrcholmi n -1 krát postupne vykonať relaxáciu všetkých orientovaných hrán grafu. Vykonávať všetkých n „fáz“ relaxácií všetkých hrán grafu nie je potrebné. Výpočet môžeme ukončiť ihneď po vykonaní takej „fázy“ relaxácií všetkých hrán grafu, v ktorej sa pre žiaden vrchol neznížil odhad ohodnotenia najlacnejšej cesty (sledu) zo štartovacieho vrcholu do tohto vrcholu.

☐ 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.

☐ Ak n je veľkosť vstupu a m je veľkosť vzorky, potom nájdenie všetkých výskytov vzorky vo vstupnom texte je možné realizovať algoritmom s časovou zložitosťou $O(n + m)$, no neexistuje žiaden algoritmus pre tento problém bežiaci v čase $O(\log^2 n + m)$.

V úlohách 9-11 budeme uvažovať nasledujúci neorientovaný ohodnotený graf:



9. (1.5b) Ak prehľadávanie do hĺbky je inicializované vo vrchole D a susedné vrcholy „spracúvame“ vždy v abecednom poradí, zapíšte postupnosť vrcholov grafu, v akej budú navštívené:

_____

10. (1b) Pre aké celočíselné hodnoty x bude hrana CD za každých podmienok vybraná Primovým aj Kruskalovým algoritmom do najlacnejšej kostry?

_____

11. (2b) Zapíšte postupnosť hrán grafu, v akej budú pridané jednotlivé hrany grafu do minimálnej kostry pri aplikovaní Primovho algoritmu pri voľbe $x=6$ s iniciálnym vrcholom A: