



Závèrečný test praktická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Doplňujúce zdrojové kódy sú na stránke predmetu PAZ1b. Funkčnosť každého riešenia musí byť preukázaná spustením na testovacom vstupe - nespustiteľné riešenia neumožňujú zisk príslušných bodov.

r PÁZdniny a akčné leto

Brigády (14 + 7 bodov, dynamické programovanie)

Mnoho študentov sa počas letných prázdnin chystá na brigády. Cieľom brigád je spoznanie nových ľudí, získanie nových vedomostí a zručností, ale často krát je to hlavne o peniazoch. Nie len informatici si kladú otázku, ako na brigádach zarobiť čo najviac peňazí. Váš spolužiak vytvoril program, ktorý z online ponuky brigád na leto stiahne informácie o každej z nich. O každej brigáde vieme dátum začiatku, dátum ukončenia a finančné ohodnotenie za celú brigádu. Napíšte program, ktorý vypočíta, koľko najviac peňazí môžeme počas leta zarobiť. Poznámka: Predpokladáme, že naraz môžeme byť len na jednej brigáde a po nástupe na brigádu je nutné ju absolvovať celú, aby sme za ňu dostali zaplatené. Výplata za brigádu je na konci jej posledného dňa.

Formát vstupu: Prvý riadok obsahuje počet brigád n . Každý z ďalších n riadkov obsahuje informácie o jednej brigáde, deň začiatku brigády (poradové číslo od začiatku leta), deň ukončenia brigády (poradové číslo od začiatku leta) a sumu zarobených peňazí.

Príklad:

```
5
2 2 30
4 6 90
8 10 90
12 12 30
5 9 250
```



Úlohy:

(14b) Implementujte program, ktorý načíta textový súbor a vypíše najviac koľko peňazí je možné zarobiť na brigádach. Akceptované sú iba riešenia v polynomiálnom čase vzhľadom na počet brigád.

(+7b) Doplňte program tak, aby vypísal takú voľbu brigád, za ktoré sa dá zarobiť najviac peňazí.

Hint: Označme si $R[d]$ maximálny zárobok, ktorý vieme získať od začiatku leta po koniec d -teho dňa leta.

Čas na hobby - vesmírne stromy (14 bodov, stromy)

Váš spolužiak Marek má konečne čas na svoje veľké hobby, ktorým je vesmír a všetko čo s ním súvisí. Preto začal študovať údaje o vesmírnych misiách. V minulosti každá krajina a každá spoločnosť, ktorá realizovala lety do vesmíru, používala vlastnú techniku, vlastný software a vlastný personál. V ére spoločných projektov je potreba zjednotiť toho čo najviac. Ako prvý sa zjednotil vzduchový uzáver, ktorý na vstupe do ISS používajú všetky zúčastnené krajiny a firmy. Podobné zjednotenie dátových štruktúr ešte nenastalo. Ved' NASA (USA), ESA (EU), FKA (Rusko), CNSA (China), ISRO (India), Space-X, Orbital ATK a mnohé ďalšie používajú vlastné systémy a vlastné formáty dát. Marek si uvedomil, že aj on vie pomôcť týmto gigantom ku zjednoteniu dát. Pred zjednotením je nutné najprv zistiť, či vôbec ide o tie isté dáta. Ved' aj jednoduchá štruktúra ako strom môže mať veľa rôznych spôsobov implementácie. Preto je nutné porovnať stromy uložené v rôznych implementáciách, či reprezentujú to isté.



Úloha: Implementujte program, ktorý načíta stromy v troch rôznych reprezentáciách a overí, či zodpovedajú identickému stromu. Navyše vieme, že každá hodnota v strome sa nachádza najviac raz. Rôzne reprezentácie:

1. Referencia na koreň stromu, ktorý je reprezentovaný triedou *Osoba* zo 4. prednášky.
2. Strom je reprezentovaný dvojicou polí, kde hodnoty na rovnakých indexoch spolu súvisia. Každý index reprezentuje jednu osobu. Prvé pole obsahuje meno osoby a druhé pole obsahuje meno rodiča danej osoby alebo `null`, ak je koreňom stromu (predpokladáme, že každá osoba má unikátne meno).
3. Pole reťazcov a matica susednosti orientovaného grafu, pričom orientovaný graf uložený v matici určuje potomkov a rodičov. Hrana ide od rodiča k potomkovi. Hodnota v poli na indexe *i* vyjadruje meno *i*-tej osoby.

Hurá na tábor (12 + 3 bodov, grafy)

Matej je každoročný účastník komunitných táborov. Rozhodol sa, že ako vysokoškolák chce vrátiť komunitě to, čo mu dala a tak bude robiť dobrovoľníka v detskom tábore. Prvá akciu, ktorú mu prideliť, je zoznamovačka. V nej je treba rozdeliť deti do dvoch skupín. Deti, ktoré idú na tábor, vyznačili, koho už poznajú. Preto sa Matej rozhodol, že ich rozdelí do dvoch skupín tak, aby sa v skupine nikto s nikým nepoznal. Po chvíli rozmýšľania si uvedomil, že na reprezentáciu týchto vzťahov sú vhodné grafy. Vygooglil si, že takéto grafy sa volajú bipartitné. Aké sú to bipartitné grafy? Graf *G* nazveme bipartitným, ak množinu jeho vrcholov $V(G)$ môžeme rozdeliť na dve disjunktné podmnožiny *X* a *Y*, t.j. $X \cap Y = \emptyset$, $X \cup Y = V(G)$, s takou vlastnosťou, že pre každú hranu grafu platí, že jeden jej koniec je v množine *X* a druhý v množine *Y*.

Úlohy:

(12b) Vytvorte program, ktorý pre zadaný neorientovaný (potenciálne aj nesúvislý) graf overí, či je tento graf bipartitný.

(+3b) Upravte program tak, aby vypísal rozdelenie detí do dvoch skupín, ak je graf bipartitný.

Hint: Nech *G* je súvislý bipartitný graf. Ak si zoberieme ľubovoľný vrchol $s \in X$, tak jeho susedia (ak nejakých má) sú z množiny *Y*. Toto pozorovanie ide rozšíriť. Ak si zoberieme ľubovoľný vrchol $s \in X$, potom vrcholy, ktorých vzdialenosť od *s* je párna, sú v množine *X* a vrcholy, ktorých vzdialenosť od *s* je nepárna, sú v množine *Y*. Čo ale s hocíjakým súvislým grafom? Nuž, ak by takýto graf mal byť bipartitný, potom po rozdelení vrcholov do množín *X* a *Y* podľa vzdialenosti od zvoleného vrcholu by malo platiť, že v grafe niet hrany, ktorá by mala oba konce buď v *X* alebo *Y*.

Festival a pokazené tlmiče (14 bodov, backtracking)

Po úspešnom skúškovom období sa študenti celého ročníka rozhodli, že spolu pôjdu na festival Snina ROCK pod kameňom. Lenže z vlakovej stanice v Snine je to na Sninské rybníky ešte kus cesty. Nevyhnutnosť nosiť stan s ďalším vybavením a proviantom počas horúcich dní nerobí cestu ľahšou. Preto sa ich dobrá spolužiačka Janka rozhodla, že ich zo stanice odvezie autom. Lenže auto má obmedzený počet miest na sedenie a kvôli pokazeným tlmičom aj obmedzenú prepravnú hmotnosť. Ako skúsený informatici sa preto rozhodli, že vedú napísať program, ktorý vypočíta, na koľko najmenej jazd sa vedú všetci prepraviť zo stanice na festival.

Úloha: V textovom súbore (formát podľa vlastného uváženia) sú zapísané maximálna hmotnosť, ktorú môže Janka v aute odvieť, počet miest na sedenie, okrem vodičky, a hmotnosti jednotlivých študentov s ich batožinou. Nájdite také rozdelenie študentov na jednotlivé jazdy auta, aby sa celkový počet jazd auta potrebných na odvezenie všetkých študentov minimalizoval.

Hint: Uvedomte si, že každý spôsob rozdelenia študentov je určený tým, v akom poradí nastúpia do auta, resp. v akom poradí sa odvezú

Túra alebo tam a späť (6 + 9 bodov, grafové algoritmy)

Niektorí radi chodíme do prírody. Či sme na festivale na Sninských Rybníkoch, v tábore alebo len máme čas medzi brigádami, vždy si nájdeme čas na dlhšiu alebo kratšiu túru. Väčšina z nás preferuje túru, kde sa nevraciamе tou istou cestou ale inou. Zoberme si mapu reprezentovanú ako neorientovaný ohodnotený graf. Ako takú túru naplánovať? Pre jednoduchosť cestu tam (zo štartu k cieľu) zoberieme ako najkratšiu cestu a naspäť (z cieľu do štartu) pôjdeme takou cestou, ktorá nemá s cestou tam žiaden spoločný vrchol ani hranu.

Úlohy:

(6b) Implementujte program, ktorý načíta graf, začiatkový a cieľový vrchol (formát podľa vlastného uváženia) a nájde (vypíše) najkratšiu cestu zo štartovacieho vrcholu do cieľového.

(+9b) Rozšírte program, aby našiel „druhú najkratšiu cestu“, kde „druhá najkratšia cesta“ nemá žiaden spoločný vrchol s najkratšou cestou okrem štartovacieho a cieľového vrcholu.

Súčtový zoznam (4+4 bodov, spájané zoznamy)

Uvažujme triedu SpajanyZoznam z prednášky o spájaných zoznamoch. Do triedy SpajanyZoznam pridajte metódu `suctovyZoznam`, ktorá vráti novovytvorený spájaný zoznam, ktorého hodnoty budú postupne súčty skupiniek za sebou idúcich k hodnôt. Ak na konci zoznamu už nezostáva k hodnôt, posledná hodnota v novom zozname bude súčtom zostávajúcich hodnôt.

```
public SpajanyZoznam suctovyZoznam(int k)
```

Príklady:

- Volanie metódy `suctovyZoznam(3)` nad zoznamom `[3, 2, 4, 1, 8, 3, 2, 5, 0]` vráti zoznam `[9, 12, 7]`.
- Volanie metódy `suctovyZoznam(3)` nad zoznamom `[8, 3, 4, 6, 8, 10, 2, 1]` vráti zoznam `[15, 24, 3]`.

Požiadavky na implementáciu:

- Pamäťová zložitosť $O(1)$
- +4 body: časová zložitosť $O(n)$ pri n -prvkovom zozname nezávislá od voľby parametra k .