Záverečný 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.

Trojice (10 bodov + 3 body za efektívnosť, backtracking)

Máme $n = 3.k$ závaží s hmotnosťami $m_1, m_2, \dots, m_n \in \mathbb{N}$. Vieme týchto n závaží rozdeliť na k trojíc tak, že každá trojica má rovnakú celkovú hmotnosť? Vytvorte program, ktorý zodpovie na takúto otázku.

PAZUber (14+10 bodov, dynamické programovanie)

Služby ako Uber na zdieľanie miest v aute už stihli vyvolať vlnu demonštrácii taxikárov. Juraj, vlastník dvojmiestneho auta, každý deň cestuje autom stálou a dlhou trasou do práce. Aby druhé (prázdne) miesto v aute mu zarobilo aspoň čiastočne na benzín, rozhodol sa používať službu PAZUber. Po prihlásení sa na PAZUber vidí Juraj požiadavky na prevoz autom na jeho trase. Kvôli jednoduchosti prepravy služba PAZUber definuje na jeho trase zastávky, kde môžu pasažieri nastúpiť alebo vystúpiť. Tie sú očíslované postupne od 1 po N podľa poradia na trase (1 je miesto, kde jeho trasa začína, a N je miesto, kde jeho trasa končí). Každá požiadavka na prepravu pasažiera je trojica:

- zastávka na trase, kde chce pasažier nastúpiť,
- zastávka na trase, kde chce pasažier vystúpiť,
- ponúknutá cena za prepravu (tú pasažier zaplatí Jurajovi za odvoz)

Zo zoznamu požiadaviek na prepravu Juraj vyberie také požiadavky pasažierov, o ktorých prepravu sa postará. Samozrejme, Jurajov záujem je maximalizovať príjem za odvozy. Chce preto zo zoznamu požiadaviek akceptovať také požiadavky tak, aby čo najviac zarobil. Keďže má iba dvojmiestne auto, v aute môže mať v každom okamihu nanajvýš jedného pasažiera.

Úloha: Zo zoznamu požiadaviek na prepravu načítaného z textového súboru (každý riadok reprezentuje jednu požiadavku ako trojicu [odkiaľ, kam, cena]) vyberte také požiadavky, aby ich Juraj mohol zrealizovať (max. 1 pasažier v každom okamihu) a zároveň Jurajov príjem bol maximálny.

Rada: Označme si $P[i]$ maximálny Jurajov príjem do okamihu, keď za nachádza na zastávke i a jeho auto je prázdne (pasažiera nemá alebo vystúpil na i-tej zastávke). Aké situácie mohli nastať na tejto zastávke? Nezabudnite, že mohlo existovať viacero požiadaviek na prepravu s cieľom v i-tej zastávke.

Hodnotenie: 14 bodov za výpočet maximálneho možného príjmu, 10 bodov za vypísanie požiadaviek, ktoré treba akceptovať, aby sa tento príjem dosiahol.

Súčtový zoznam (3+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.

Bipartitné grafy (12 bodov, grafy)

Zaujímavou skupinou grafov (aj z pohľadu praxe) sú takzvané bipartitné grafy. Aké to sú 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.

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

Rada: 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 hocijakým 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.

Návraty k midtermu (5 bodov + 2 body za efektívnosť)

Z teoretického polsemestrálneho testu vieme, že platí toto tvrdenie: *„Uvažujme 2 polia dĺžky n. V čase $O(n.\log n)$ môžeme zistiť, či existuje hodnota, ktorá sa nachádza v oboch poliach.“*

Naprogramujte metódu, ktorá takýto test v uvedenom čase realizuje.

```
public boolean spolocnaHodnota(int[] p1, int[] p2)
```