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.

Riadkovo-stĺpcové súčty (8+6 bodov, backtracking)

Uvažujme tabuľku s rozmermi 4×6 vyplnenú hodnotami 0 a 1. Pre každý riadok a stĺpec tabuľky máme spočítaný počet jednotiek v danom riadku, resp. stĺpci:

0	1	0	1	0	1	3
0	0	1	0	1	0	2
1	0	0	1	0	1	3
0	1	0	0	0	0	1
1	2	1	2	1	2	9

Na tieto riadkovo-stĺpcové súčty môžeme pozerať ako na akúsi najjednoduchšiu formu kontrolných súčtov. Ak by sa zmenil obsah (z 0 na 1 alebo naopak) nanajvýš jednej bunky, vďaka takýmto kontrolným súčtom dokážeme veľmi rýchlo identifikovať zmenu v tabuľke. Nás bude zaujímať táto otázka: Do akej miery riadkovo-stĺpcové súčty určujú/obmedzujú obsah tabuľky?

Vytvorte program, ktorý pre zadané hodnoty riadkovo-stĺpcových súčtov nájde nejaké vyhovujúce vyplnenie tabuľky (ak existuje) s rozmermi $m \times n$ hodnotami 0/1. Vstup načítajte z textového súboru.

Formát vstupu: V prvom riadku sú rozmery tabuľky m a n, v druhom riadku je m riadkových súčtov a v ďalšom riadku n stĺpcových súčtov. Príklad:

```
4 6
3 2 3 1
1 2 1 2 1 2
```

Rada: Každé dvojrozmerné pole s rozmermi $m \times n$ je možné uložiť do jednorozmerného poľa s dĺžkou $m.n$ (napr. po riadkoch alebo stĺpcoch).

Náročnejšia verzia (+6 bodov): Riešte tú istú úlohu ale s tým rozdielom, že obsah niektorých buniek je už vopred určený. Obsah tabuľky načítajte z textového súboru, pričom hodnota 2 kóduje, že ide o bunku tabuľky bez vopred určenej hodnoty (základná verzia úlohy zodpovedá situácii, keď žiadna z buniek tabuľky nemá vopred určenú hodnotu).

Príklad:

```
4 6
3 2 3 1
1 2 1 2 1 2
0 1 0 1 2 1
0 2 2 0 2 0
1 2 2 1 2 1
0 1 0 0 2 0
```

Návraty k midtermu (5 bodov)

Vieme, že platí toto tvrdenie: **„***Máme 2 usporiadané (neklesajúce) postupnosti (polia)* A *veľkosti* n *a* B *veľkosti* m*. Potom v čase* $O(n+m)$ *možno zistiť počet rôznych hodnôt s tou vlastnosťou, že sa nachádzajú v oboch postupnostiach, t.j. aj v* A *aj v* B*.*“

Naprogramujte metódu, ktorá v uvedenom čase zistí počet rôznych hodnôt, ktoré sa nachádzajú v oboch poliach. Metóda nesmie modifikovať referencované polia. Akceptovaná pamäťová zložitosť je $O(1)$.

```
public int velkostPrieniku(int[] a, int[] b)
```

Príklad: $a = [1, 1, 2, 5, 5, 5, 9, 17, 20]$, $b = [1, 4, 5, 5, 8, 9, 9, 21, 21, 21]$ -> 3

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 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 v okamihu, keď za nachádza na zastávke i a jeho auto je prázdne (pasažiera nemá alebo vystúpil na i-tej zastávke). Uvažujme taký výber požiadaviek na prepravu, ktorý maximalizuje $P[i]$. V tomto výbere na i-tej zastávke mohli nastať dve situácie. (1) Juraj na tejto zastávke „nevyložil“ pasažiera a teda jeho príjem je rovnaký ako na predchádzajúcej zastávke. (2) Juraj na tejto zastávke „vyložil“ nejakého pasažiera, čím zvýšil svoj príjem v porovnaní so zastávkou, kde tento pasažier nastúpil.

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

Stromovač (4+8 bodov, stromy)

Uvažujme triedu `Osoba` z prednášky o stromoch:

```
public class Osoba {
	private String meno;
	private List<Osoba> deti = new ArrayList<Osoba>();

	public Osoba(String meno) {
		this.meno = meno;
	}

	public void pridajDieta(Osoba dieta) {
		deti.add(dieta);
	}

	public void doSuboru(File suborSPopisomStromu) {

	}

	public static Osoba zoSuboru(File suborSPopisomStromu) {

	}

}
```

Implementujte metódy `doSuboru` a `zoSuboru`, ktoré realizujú uloženie obsahu stromu do textového súboru, resp. vytvorenie stromu podľa obsahu textového súboru. Formát súboru si zvoľte podľa vlastného uváženia – v prípade oboch metód však musí byť rovnaký.

Upresnenie: Inštančná metóda `doSuboru` uloží do zadaného textového súboru popis stromu, ktorého koreňom je objekt triedy `Osoba`, nad ktorým sa táto metóda volá. Statická metóda `zoSuboru` prečíta zo zadaného textového súboru popis stromu, podľa tohto popisu ho zrekonštruuje a vráti referenciu na objekt triedy `Osoba`, ktorý je koreňom vytvoreného stromu.

Deň D (4+8 bodov, grafové algoritmy)

Súostrovie PAZ Oceánia sa skladá z n ostrovov číslovaných $0 \dots n-1$. PAZ Oceánia získala grant na vybudovanie pravidelných obojsmerných lodných spojení medzi ostrovmi. Už je známy aj harmonogram, v akom sa jednotlivé spojenia budú postupne otvárať. Každý deň je v pláne otvoriť práve jedno nové (obojsmerné) lodné spojenie. Obyvatelia PAZ Oceánie sa už nevedia dočkať dňa, kedy sa bude dať cestovať medzi ľubovoľnými dvoma ostrovmi súostrovia.

Zadanie: V textovom súbore je zadané číslo n - počet ostrovov PAZ Oceánie. V každom z ďalších riadkov sa nachádza dvojica medzerou oddelených čísel x a y, ktoré označujú vytvorené obojsmerné lodné spojenie medzi ostrovmi s číslami x a y. Spojenia v súbore sú uložené chronologicky, t.j. podľa času, kedy sa uvedú do prevádzky. Vytvorte program, ktorý vráti, v koľký deň podľa harmonogramu sa bude dať po prvý krát cestovať medzi ľubovoľnými dvoma ostrovmi súostrovia (potenciálne aj s prestupmi).

Bodovanie: 4b za riešenie v polynomiálnom čase + 8b za riešenie v čase $O(n^2)$.

Rada: Inšpirujte sa grafovými algoritmami, ktoré sú založené na postupnom pridávaní hrán do grafu a testovaní súvislosti.