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.

Návraty k midtermu (5 bodov)

Vieme, že platí toto tvrdenie: **„***V čase $O(n.logn)$ môžeme zistiť, koľko hodnôt sa v n-prvkovom poli vyskytuje práve raz.*“

Naprogramujte metódu, ktorá takýto výpočet v uvedenom (aj priemernom) čase realizuje.

```
public int pocetJedinecnych(int[] p)
```

Turné po Európe (15+3 bodov, backtracking)

Nezávislá košická skupina P.A.Z. (a.k.a. Kraftwerk revival) sa teší obrovskej popularite a preto plánuje vyraziť na veľkolepé európske koncertné turné. Počas neho chcú samozrejme potešiť čo najviac svojich fanúšikov, čo znamená navštíviť čo najviac miest. V každom meste môžu odohrať nanajvýš jeden koncert (pre prípad, že by koncert skončil fiaskom). Ich turné začína a končí v Košiciach. Medzi jednotlivými mestami cestujú vlastným minibusom. Každý presun medzi mestami niečo stojí (nafta, mýtne poplatky, atď.). Ako nezávislí hudobníci nemajú peniaze, preto tieto náklady chcú platiť len z toho, čo zarobili na už uskutočnených koncertoch v rámci turné. Z tohto dôvodu si musia celé turné (navštívené mestá a poradie, v akom ich navštívia) veľmi starostlivo naplánovať.

Zdroje informácií:

- Pre každú dvojicu miest poznáme celkové náklady na presun medzi týmito mestami (napr. Bratislava-Viedeň 80 EUR, Budapešť-Košice 420 EUR, Bratislava-Ostrava 110 EUR, atď.)
- Honorár, resp. zisk z koncertu v danom meste (napr. Bratislava 200 EUR, Košice 150 EUR)

Zadanie: Načítajte si vstupy (zdroje informácií) vo vhodnom formáte (mestá môžete stotožniť aj s číslami, t.j. môžete si mestá očíslovať) a nájdite najdlhšiu realizovateľnú postupnosť koncertných miest. Realizovateľnú znamená, že peniaze, ktoré majú k dispozícii po koncerte v danom meste sú dostatočné na to, aby sa dostali do ďalšieho mesta v rámci turné. Postupnosť začína a končí v Košiciach. Zároveň zisk z košického koncertu predstavuje ich prvé peniaze, z ktorých musia zafinancovať cestu do ďalšieho koncertného mesta.

Route 66 (14+6 bodov, dynamické programovanie)

Prázdniny sú čochvíľa tu. Už len spraviť záverečný test z PAZ1b na dostatočný počet bodov a potom dlhá cesta naprieč USA – po legendárnej Route 66. Takúto dlhú cestu si však treba dôsledne naplánovať. Popri ceste je n hotelov na kilometrovníkoch $a_1 < a_2 < \cdots < a_n$ (cesta začína na kilometri 0, kilometrovníky sa rátajú ako vzdialenosti od začiatku cesty). Tieto miesta sú jediné miesta vhodné na prenocovanie. Je však na nás, ktoré z nich si vyberieme ako miesta na prenocovanie. Naša cesta končí v poslednom n-tom hoteli, takže v ňom sa určité zastavíme. Zo skúsenosti je najlepšia vzdialenosť prejdená za deň asi 300 km – vtedy si

cestovateľ cestu a zaujímavosti v okolí najviac užije. No hotely nie sú rozmiestnené každých 300 km – niekedy musíme medzi 2 hotelmi, v ktorých nocujeme, prejsť viac, inokedy menej ako 300 km. Čím viac sa počet prejdených kilometrov líši od hodnoty 300, tým viac sme nespokojnejší. Matematicky môžeme cestovateľskú nespokojnosť v daný deň vyjadriť ako $(300 - x)^2$, kde x je počet kilometrov prejdených v tento deň. Našťastie si ako znalci algoritmov vieme napísať program, ktorý nám pre zadané umiestnenia hotelov určí také hotely na prenocovanie počas cesty, že celkový súčet cestovateľských nespokojností počas všetkých dní bude najmenší možný.

Rada: Označme si $P[i]$ najmenšiu celkovú cestovateľskú nespokojnosť na ceste, ktorá končí v i-tom hoteli.

Hodnotenie:

- 14 bodov za nájdenie najmenšej dosiahnuteľnej celkovej cestovateľskej nespokojnosti
- 6 bodov za nájdenie hotelov, v ktorých sa máme pri optimálnej ceste zastaviť

Stromovač (12 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 vypisStrom(int uroven) {
		for (int i = 0; i < uroven; i++) {
			System.out.print('*');
		}
		System.out.println(meno);
		for (Osoba dieta : deti) {
			dieta.vypisStrom(uroven + 1);
		}
	}
}
```

Ak zavoláme metódu `vypisStrom(0)` na koreni stromu, dostaneme textový zápis obsahu tohto stromu. Je to teda spôsob ako nejaký všeobecný strom zakódovať.

Do triedy `Osoba` pridajte metódu `vytvorStrom`, ktorá vráti referenciu na koreň novovytvoreného stromu podľa zadaného textového súboru. Vytvorený strom má mať tú vlastnosť, že ak na jeho koreni zavoláme metódu `vypisStrom(0)` dostaneme výpis totožný s obsahom tohto textového súboru. Môžete predpokladať, že súbor obsahuje korektný vstup.

```
public static Osoba vytvorStrom(File suborSPopisomStromu)
```

Bezpečné cesty (12+7 bodov, grafové algoritmy)

Dopravná firma potrebuje previesť nebezpečný náklad. Ak takýto náklad chce previesť cez CHKO alebo inú chránenú oblasť, potrebuje získať špeciálne povolenia. Je preto lepšie sa cestám vedúcim cez takéto problematické oblasti vyhnúť – aj na úkor dlhšej cesty.

Zadanie: V textovom súbore (formát podľa vlastného uváženia) je uložený ohodnotený graf popisujúci cestnú sieť. Všetky spojnice (hrany) miest (vrcholy) sú obojsmerné. O každej spojnici (hrane) poznáme jej dĺžku a vieme, či táto spojnica prechádza cez CHKO. Vytvorte program, ktorý pre zadané štartovacie a cieľové mesto nájde takú cestu medzi nimi, ktorá obsahuje čo najmenej spojníc prechádzajúcich cez CHKO. Ak je takých ciest viac, program nech vráti najkratšiu z nich.

Rada: Ako ohodnotenie cesty neuvažujte len jedno číslo h vyjadrujúce počet hrán idúcich cez CHKO, ale usporiadanú dvojicu (h, d), kde h je počet hrán na tejto ceste prechádzajúcich cez CHKO a d je súčet ohodnotení hrán na tejto ceste. Odhady dĺžky najlacnejších ciest musia byť preto taktiež dvojice, ktoré porovnávame lexikograficky. Pritom platí, že $(h_1, d_1) < (h_2, d_2) \Leftrightarrow h_1 < h_2 \vee (h_1 = h_2 \wedge d_1 < d_2)$.

Bodovanie: 12 b za dĺžku najkratšej cesty spolu s počtom hrán na nej, 7 b za jej výpis.