# 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.

## **PAZ1b a grafy** (13 bodov, backtracking)

Problém 3-zafarbiteľnosti grafu je veľmi populárny grafársko-algoritmický problém. Cieľom je pre zadaný neorientovaný graf zistiť, či je možné každý jeho vrchol zafarbiť jednou z 3 farieb tak, aby žiadne 2 susedné vrcholy nemali rovnakú farbu. Skôr či neskôr počas svojho štúdia natrafíte na dôkaz toho, že tento problém je *NP-úplný*, t.j., nie je možné ho riešiť v čase $O(n^p)$ kde $p$ je konštanta (za predpokladu, že P != NP). Avšak už teraz viete napísať program, ktorý takýto test 3-zafarbiteľnosti grafu spraví v exponenciálnom čase.

Vytvorte program, ktorý načíta graf z textového súboru (formát súboru je na vás) a vypíše, či zadaný graf je 3-zafarbiteľný.

## **PAZ1b a ekonómia** (12 bodov, grafové algoritmy)

Euro prežíva krízu. Dolárový trh tiež nie je žiadna sláva. Čo ak náš menový systém padne a vrátime sa k bártrovému (výmennému) obchodu? Študenti EFM musia byť pripravení aj na takúto eventualitu... Ak sa tak stane, dá sa očakávať, že vznikne akási komoditná burza. V nej každý deň obchodníci prídu s množstvom ponúk typu: "Vymením 3 kilá cukru (chcem) za 5 kíl múky (mám)." Každú takúto ponuku vieme zapísať jedným riadkom v textovom súbore:

```
3 cukor 5 muka
```

Samozrejme obchodníkov ponúkajúcich výmenu cukru za múku môže byť viacero a môžu ponúkať rôzne „kurzy výmeny". Na komoditnej burze tiež predpokladáme, že obchodník je pripravený nakúpiť neobmedzené množstvo požadovanej komodity (cukor) a disponuje neobmedzeným množstvom ponúkanej komodity (múka).

Niekedy je obchodovanie komplikovanejšie. Napríklad máte cukor a chcete si kúpiť voľné minúty u mobilného operátora. No žiaden operátor nechce cukor za voľné minúty. Môže sa však stať, že máte na burze aj takéto dve ponuky:

```
3 cukor 5 muka
2 muka 30 minut
```

Vtedy môžete najprv cukor zameniť za múku a následne si múku zameníte za voľné minúty. Takáto „reťaz" postupných zámen môže byť aj oveľa dlhšia.

Vytvorte program, ktorý načíta zoznam ponúk na burze. Potom pre zadané dve komodity určí, či je možné zrealizovať postupnosťou výmen výmenu prvej komodity (napr. cukor) za druhú komoditu (napr. voľné minúty).

## **PAZ1b a algebra** (15 bodov, dynamické programovanie)

Spomínate si na násobenie matíc? Ak $A$ je matica typu $r_A \times s_A$ ($r_A$ riadkov a $s_A$ stĺpcov) a $B$ je matica typu $r_B \times s_B$ tak na to, aby sme ich mohli vynásobiť musí platiť, že $s_A = r_B$. Výsledkom násobenia je matica $C$ typu $r_A \times s_B$. Na to, aby sme vypočítali maticu $C$, potrebujeme celkom $r_A \cdot s_A \cdot s_B$ operácií násobenia prvkov matíc (použijúc naivný školský algoritmus). Teda ak násobíme maticu typu $5 \times 3$ a maticu typu $3 \times 8$, výsledkom je matica typu $5 \times 8$ a na jej výpočet naivným (školským) algoritmom potrebujeme $5 \cdot 3 \cdot 8 = 120$ operácií násobenia prvkov matíc. Z algebry viete, že operácia násobenia matíc je asociatívna, t.j. nezáleží na ozátvorkovaní. A to ide celkom šikovne využiť. (Pripomíname, že násobenie matíc nie je komutatívne.)

Zoberme si 3 matice: $A_1$ typu $5 \times 4$, $A_2$ typu $4 \times 6$ a $A_3$ typu $6 \times 2$. Výsledkom súčinu $A_1 A_2 A_3$ je matica typu $5 \times 2$. Ak by sme matice násobili ako $(A_1 A_2) A_3$, na výpočet $A_1 A_2$ potrebujeme celkom $120$ $(= 5 \cdot 4 \cdot 6)$ násobení. Potom na vynásobenie matíc $(A_1 A_2)$ a $A_3$ potrebujeme celkom $60$ $(= 5 \cdot 6 \cdot 2)$ násobení. Celkom teda máme $120 + 60 = 180$ násobení. Ak by sme súčin $A_1 A_2 A_3$ počítali ako $A_1(A_2 A_3)$, postačí nám už len $88$ $(= 4 \cdot 6 \cdot 2 + 5 \cdot 4 \cdot 2)$ násobení. Najprv vypočítame $A_2 A_3$ a potom súčin $A_1(A_2 A_3)$.

Vidíme, že na ozátvorkovaní záleží. Problém je jasný: Nájsť také ozátvorkovanie súčinu matíc, aby sme výslednú maticu vypočítali s čo najmenším počtom operácií násobenia. Vytvorte program bežiaci v polynomiálnom čase, ktorý pre zadanú postupnosť matíc $A_1, A_2, \dots, A_n$ nájde minimálny počet operácií násobenia potrebných na výpočet ich súčinu $A_1 A_2 \dots A_n$. Keďže za sebou idúce matice musia byť kompatibilné, celý vstup vieme popísať pomocou $n+1$ čísel $r_1, r_2, \dots, r_n, r_{n+1}$ pričom matica $A_i$ je typu $r_i \times r_{i+1}$.

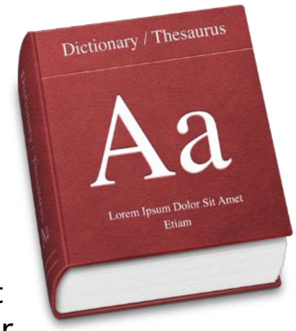
**Návod**: Označme si $M[i,j]$ minimálny počet operácií násobenia na výpočet súčinu matíc $A_i A_{i+1} \dots A_j$. Zjavne $M[i,i] = 0$. Podobne $M[i,i+1] = r_i \cdot r_{i+1} \cdot r_{i+2}$. Zoberme si teraz $M[i,j]$ také, že $j - i > 1$. Keďže operácia súčinu matíc je binárna, súčin $A_i A_{i+1} \dots A_j$ musel vzniknúť ako súčin $(A_i A_{i+1} \dots A_k)(A_{k+1} A_{k+2} \dots A_j)$ pre nejaké $k$. Ak chceme minimalizovať počet operácií násobenia pri rozdelení súčinu za k-tou maticou, dostaneme, že počet potrebných operácií násobenia je $M[i,k] + M[k+1,j] + r_i \cdot r_{k+1} \cdot r_{j+1}$. Keďže nevieme, ktoré $k$ je „najlepšie“, tak, ako je to v dynamickom programovaní zvykom, vyskúšame všetky $k$ a vyberieme to „najlepšie“. Preto:

$$M[i,j] = \min \{M[i,k] + M[k+1,j] + r_i \cdot r_{k+1} \cdot r_{j+1} | i \leq k < j\}$$

## **PAZ1b a biológia** (12 bodov, greedy algoritmy)

Neoddeliteľnou časťou práce biológa je práca v teréne. Zaujímavé lokality sú zvyčajne dosť vzdialené od pracoviska/školy a tak na tieto lokality je treba robiť niekoľkodňové „výpravy“. S každou takouto výpravou je spojené množstvo príprav a tak sa vedenie *Ústavu biologických a ekologických vied* rozhodlo organizovať presne týždňové výpravy (terénne cvičenia a pozorovania). Pre hlavnú tohtoročnú pozorovaciu lokalitu majú na základe prognóz a dlhoročných pozorovaní zoznam dátumov, kedy sa určite oplatí navštíviť túto lokalitu. Tieto dátumy sa nachádzajú v textovom súbore (kvôli jednoduchosti môžete predpokladať, že dátum je zadaný ako poradové číslo dňa v roku, t.j. deň č. 1 je 1.1.2012, deň č. 2 je 2.1.2012, atď.). Pozor, tieto dátumy nemusia byť usporiadané. Vypočítajte minimálny počet týždňových výprav, ktoré sú potrebné na to, aby bolo možné stráviť v pozorovacej lokalite každý deň, kedy sa oplatí túto lokalitu navštíviť (je v zozname dátumov v textovom súbore). Očakáva sa riešenie bežiace v polynomiálnom čase.

**PAZ1b a jazyky** (13 bodov)




Tak ako je pre informatika dôležitý počítač, tak je pre jazykára dôležitý slovník – a nie jeden. Taký slovník obsahuje kopu slov, ktoré sú zvyčajne usporiadané abecedne (lexikograficky). Keď jazykári porovnávajú a analyzujú jazyky, neraz sa zvyknú zamýšľať nad podobnosťou týchto jazykov. Dá sa čakať, že čím sú jazyky podobnejšie, tým viac rovnakých alebo podobných slov obsahujú. Moderní jazykári využívajú moderné nástroje – napríklad počítače. Aj slovníky majú uložené elektronicky. Trebárs súbor `slovencina.txt` obsahuje zoznam spisovných slov slovenského jazyka, súbor `cestina.txt` zoznam spisovných slov češtiny, atď. V každom riadku je jedno slovo. Slová sú v textovom súbore (ako sa na slovník patrí) usporiadané abecedne. Jazykárom by pomohol program, ktorý zoberie názvy dvoch slovníkových súborov a zráta počet slov, ktoré sa nachádzajú v oboch slovníkoch (napr. slovo informatika sa istotne nachádza v oboch slovníkoch: v slovníku `slovencina.txt` aj `cestina.txt`). **Pozor:** slovníkové súbory môžu byť obrovské a obsahovať obrovské počty slov. Dokonca také veľké, že sa nezmestia ani do pamäte počítača. Vytvorte program, ktorý tento problém vyrieši, t.j. zráta počet rovnakých slov v dvoch textových súboroch. Ak $n$ je počet slov v prvom slovníku a $m$ je počet slov v druhom slovníku, očakáva sa od vás riešenie bežiace v čase $O(n + m)$.