



## 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.

### Domček jedným ťahom (17 bodov, backtracking alebo ...)

Na vstupe je útvar, ktorý môžeme reprezentovať ako súvislý neorientovaný neohodnotený graf. Vytvorte program, ktorý po načítaní daného grafu (útvaru) a štartovacieho vrcholu rozhodne, či je možné daný útvar nakresliť jedným ťahom tak, že začneme v štartovacom vrchole. Ak daný útvar je možné nakresliť jedným ťahom so začiatkom v štartovacom vrchole, tak nech program vypíše postupnosť hrán tak, ako by sme mali daný útvar nakresliť jedným ťahom.

| Príklad vstupu:  | Komentár:  | Útvar: |
|--|--|--------|
| 5 8<br>0<br>0 1<br>0 3<br>0 2<br>1 3<br>1 2<br>2 3<br>2 4<br>3 4 | Máme 5 vrcholov, 8 hrán.<br>Chceme začať vo vrchole 0.<br>Hrana spájajúca vrcholy 0 a 1.<br>Hrana spájajúca vrcholy 0 a 3.<br>Hrana spájajúca vrcholy 0 a 2.<br>Hrana spájajúca vrcholy 1 a 3.<br>Hrana spájajúca vrcholy 1 a 2.<br>Hrana spájajúca vrcholy 2 a 3.<br>Hrana spájajúca vrcholy 2 a 4.<br>Hrana spájajúca vrcholy 3 a 4. |        |

**Rada 1:** V grafárskej terminológii hľadáme Eulerovský ťah. Keďže Eulerovský ťah obsahuje všetky hrany grafu, na Eulerovský ťah sa môžeme pozerat' aj ako na akúsi „špeciálnu“ permutáciu hrán grafu s vlastnosťou, že hrany na seba nadväzujú.

**Rada 2:** Použitie matice susednosti alebo knižnice PAZGraphs je možné, ale pri niektorých prístupoch k vyriešeniu úlohy môže veci skomplikovať.

**Komentár:** Úlohu stačí vyriešiť backtrackingom (a teda v čase  $O(m!)$ ). Aplikovaním poznatkov z teórie grafov však túto úlohu ide vyriešiť aj v lineárnom čase.

### Minimálna triangulácia konvexného mnohouholníka (18 bodov, dynamické programovanie)

Na internete sme našli návod, ako nájsť minimálnu trianguláciu konvexného mnohouholníka (v prílohe zadania). Aplikovaním dynamického programovania implementujte riešenie tohto problému na základe tohto návodu.

### PAZAir 2 (15 bodov, grafové algoritmy)

Letecká spoločnosť PAZAir prevádzkuje okrem pravidelných leteckých liniek aj individuálne lety malými lietadlami. Klient si pri individuálnom lete objednáva dopravu zo zvoleného štartovacieho letiska do zvoleného cieľového letiska. Pri dlhších letoch vzhľadom na malý objem palivovej nádrže malých lietadiel vzniká potreba naplánovať aj medzipristátia za účelom doplnenia paliva.

Fakty:

- Lietadlo má spotrebu  $S$  litrov na 100 km letu, pristávanie spotrebuje navyše  $P$  litrov paliva a vzlet  $V$  litrov paliva. Kapacita nádrže lietadla je  $K$  litrov paliva.
- Každé letisko má svoju geografickú polohu. Pre jednoduchosť budeme predpokladať, že geografická poloha letiska je zadaná ako  $x$ -ová a  $y$ -ová súradnica bodu v rovine.

**Úloha:** Pre zadané štartovacie a cieľové letisko nájdite najkratšiu realizovateľnú trasu letu zo štartovacieho do cieľového letiska zohľadňujúcu obmedzený objem palivovej nádrže.

Formát vstupu v textovom súbore si zvolte tak, aby bol pre vás čo najviac vyhovujúci. Odporúčame mať v prvom riadku súboru všeobecné parametre ( $S$ ,  $P$ ,  $V$ ,  $K$ ), počet letísk  $n$  (napr. číslovaných od  $0, \dots, n - 1$ ), číslo štartovacieho a číslo cieľového letiska. Potom v ďalších  $n$  riadkoch mať v každom riadku 2 čísla reprezentujúce pozíciu (súradnice) letiska na mape.

## Chvost s ohraničeným súčtom (6 bodov, spájané zoznamy)

Uvažujme triedu `SpajanyZoznam` z prednášky o spájaných zoznamoch. Navyše predpokladajme, že zoznam môže obsahovať len kladné čísla. Do triedy `SpajanyZoznam` pridajte metódu `chvost`, ktorá zo začiatku spájaného zoznamu odstráni čo najmenej prvkov zoznamu tak, aby pre zostávajúce prvky zoznamu platilo, že ich súčet je menší alebo rovný ako zadaná hodnota (parameter `sucet`). Metóda nech pracuje v lineárnom čase vzhľadom k dĺžke zoznamu (odporúčané sú 2 prechody zoznamom).

```
public void chvost(int sucet)
```

*Príklad:* Predpokladajme, že spájaný zoznam obsahuje hodnoty `[8, 5, 2, 5, 1, 4]`. Volanie metódy `chvost(14)` spôsobí, že spájaný zoznam bude obsahovať hodnoty `[2, 5, 1, 4]`. Súčet týchto hodnôt je 12, čo je menej ako 14.

*Alternatívny pohľad na zadanie:* Chceme, aby z pôvodného zoznamu zostal najdlhší jeho možný sufix, ktorého súčet je menší ako zadaná hodnota.