# 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.

## DataBig j.s.a. – príbeh začína

Študenti PF UPJŠ sa rozhodli po motivujúcom príhovore dekana na tohtoročnom ScienceFest-e založiť start-up **DataBig** špecializovaný na oblasť analýzy veľkých dát – big data. Samozrejme nebolo to len o dekanovi, ale aj o možnosti využiť obrovský know-how naakumulovaný v rámci projektu IT Akadémia, ktorého nositeľmi sa stali Ľubo, Juraj a Erik – riešitelia projektu. Príbeh **DataBig**u nájdete v tomto zadaní...

## DataBig a Orange (16 bodov, backtracking)


DataBig dostal od spoločnosti Orange prvú zákazku: identifikovať skupinky navzájom si volajúcich osôb – teda množiny osôb, ktoré si navzájom volajú. Týmto skupinkám by operátor ponúkol zvýhodnené programy (rozšírenia programu *Navzájom zadarmo)*. Big data znamená často big problém. A tak sa rozhodli začať s jednoduchším problémom. Pre zadané číslo $k$ chcú zistiť, či medzi zákazníkmi operátora je skupinka navzájom si volajúcich osôb veľkosti aspoň $k$ (poznámka: ak existuje skupinka veľkosti $10$, existuje aj skupinka menšej veľkosti $9$, $8$, atď.). Tento problém môžeme vyjadriť v reči teórie grafov takto: Nech $n$ je počet zákazníkov. Uvažujme $n$-vrcholový jednoduchý graf $G$, v ktorom je hrana medzi vrcholmi $u$ a $v$ práve vtedy, ak osoba $u$ častejšie komunikuje s osobou $v$. Skupinka navzájom si volajúcich veľkosti $k$ zodpovedná $k$-vrcholovému podgrafu grafu G, v ktorom je medzi každými dvoma vrcholmi hrana – tento podgraf je vlastne kompletný (pod)graf veľkosti $k$.

**Úloha:** Implementujte program, ktorý načíta graf z textového súboru a pre zadané $k$ overí, či graf obsahuje kompletný podgraf veľkosti $k$.

**Hodnotenie**: 8-16 bodov podľa efektívnosti. Algoritmus je tým efektívnejší, či viac eliminujete výpočty, ktoré nevedú k prípustnému riešeniu.

## **DataBig a preklady** (15 bodov, grafové algoritmy)


Študentský start-up DataBig sa rozhodol expandovať na globálny trh. Na to je nevyhnutné webovú stránku, ale aj informácie o produktoch a službách preložiť do čo najviac jazykov. Preklady však nie sú vôbec lacná záležitosť a ako start-up majú len veľmi okresaný budget (rozpočet), keďže rokovania s Kiskom juniorom ešte nie sú dokončené. Keď už sú experti na big data, určite si poradia aj so small data. Začali googliť a vytvorili si textový súbor, ktorý v každom riadku obsahuje cenu za preklad normostrany medzi nejakými 2 jazykmi:

```
SPA GRE 52.50
```

Uvedený riadok hovorí, že za preklad jednej normostrany zo španielčiny do gréčtiny alebo naopak zaplatíme 52.50 EUR. Každý prekladateľ prekladá v oboch smeroch a za rovnakú cenu. Pre každú dvojicu jazykov máme v súbore najviac jeden riadok – ak totiž viacero prekladateľov prekladá medzi zadanými jazykmi, vyberieme si toho, ktorý ma najnižšiu cenu. Najviac jeden riadok preto, že sa môže stať, že sa pre nejakú dvojicu jazykov nenašiel žiaden prekladateľ. Ak však text nevieme preložiť priamo, môžeme využiť preklad do medzijazykov a tým aj eventuálne ušetriť. Napríklad namiesto priameho prekladu zo španielčiny do gréčtiny, môžeme text preložiť do angličtiny a potom text v angličtine do gréčtiny. Kvôli zjednodušeniu predpokladáme, že text veľkosti jednej normostrany zaberá v každom inom jazyku taktiež jednu normostranu.

**Úlohy:**

- (3b) Pre zadaný východiskový jazyk nájdite všetky jazyky, do ktorých vieme texty preložiť.
- (8b) Predpokladajme, že z východiskového jazyka vieme (eventuálne s využitím postupnosti medziprekladov do iných jazykov) texty preložiť do všetkých iných jazykov. Za akú minimálnu sumu ide jednu normostranu preložiť do všetkých uvažovaných jazykov? Upozorňujeme, že nemá zmysel text prekladať do nejakého jazyka viackrát.
- (4b) Nájdite postupnosť realizácii prekladov, ktorá zabezpečí, že sa získajú preklady normostrany vo východiskovom jazyku do všetkých uvažovaných jazykov za minimálnu cenu.

Očakáva sa riešenie bežiace v polynomiálnom čase, pričom vstupy sa načítavajú zo súboru.

**Doplnenie**: Ak to bude pre vás jednoduchšie, môžete namiesto kódov jazykov použiť čísla (napr. $0, 1, 2, 3, \ldots$).

## **DataBig a ZEON** (15 bodov, dynamické programovanie)


Reklamná kampaň ZEONu k produktu *Smart domov* získala veľký ohlas. ZEON sa však trochu prerátal, keď kampaň postavil na susedskej závisti. Ukázalo sa, že ľudia chcú ich produkt len v prípade, že ich susedia tento produkt nebudú mať. Navyše nie všetci klienti sú pre ZEON rovnocenní – kým jeden chce vybavenie a senzory za 500 EUR, iný chce vybavenie a senzory za 2000 EUR (väčší dom, chce byť viac cool, atď). Je jasné, že klient s väčšou objednávkou je pre ZEON zaujímavejší. Kampaň bola taká úspešná, že všetci obyvatelia SmartStreet v SmartCity prejavili záujem o produkt *Smart domov*. Obchodní zástupcovia ZEONu spravili predobjednávky. No v každej predobjednávke je klauzula, že žiaden zo susedov nemôže mať produkt *Smart domov* po dobu najbližších 24 mesiacov (tzv. garantovaná exkluzita) – inak je predobjednávka zo strany zákazníka neplatná. Manažéri ZEONu si teraz lámu hlavu, ktorým objednávkam vyhovieť, aby sa maximalizoval zisk? S touto úlohou oslovili start-up DataBig. (SmartStreet je obrovská ulica a domov je tam toľko, že ide už ide o big data).

**Úloha:** Na vstupe máme pole veľkosti $n$, ktorého $i$-te políčko vyjadruje celkovú hodnotu objednávky zákazníka na SmartStreet v dome s orientačným číslom $i$ (v poradí $i$-ty dom na ulici). V súlade s klauzulou v predobjednávke, ak ZEON akceptuje zákazníka s číslom $i$, nemôže akceptovať jeho susedov, t.j. zákazníkov $i-1$ a $i+1$ (prvý a posledný dom na ulici majú len jedného suseda). Nájdite taký výber zákazníkov, ktorý maximalizuje príjem ZEONu. Očakáva sa riešenie bežiace v polynomiálnom, ideálne v lineárnom čase.

**Hodnotenie**:

- 8-10 bodov za výpočet celkového príjmu v optimálnom výbere (v závislosti od efektívnosti riešenia)
- 5 bodov za vypísanie optimálneho výberu

## **DataBig a midterm** (5 bodov)

Programátori start-upu DataBig ako absolventi predmetu PAZ1b aj po čase sledujú novinky na predmete. Po jednom z midtermom si uvedomili, že platí toto tvrdenie: „*V čase* $O(n.logn)$ *môžeme zistiť, počet rôznych hodnôt v poli.*“ V rámci zlepšovania svojich programátorských skillsov sa rozhodli si naprogramovať metódu, ktorá to v tomto čase realizuje.

**Úloha:** Naprogramujte metódu, ktorá v uvedenom (aj priemernom) čase nájde počet rôznych hodnôt v poli referencovanom parametrom p. Metóda nesmie modifikovať referencované pole p. Akceptovaná pamäťová zložitosť je $O(n)$. Pri riešení nie je dovolené využívať rôzne implementácie rozhrania Set.

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

## **DataBig a fault-tolerance** (18 bodov, stromy)


DataBig dostal od zákazníka (kvôli NDA ho nemôžno menovať) požiadavku na navrhnutie fault-tolerance (chybe odolného) riešenia na spracovanie údajov uložených v binárnych vyhľadávacích stromoch. Fault-tolerance sa rozhodli dosahovať tak, že použijú súbežne rôzne implementácie algoritmov, ktoré binárne vyhľadávacie stromy nezávisle od seba naplnia. Potom overia, či hodnoty uložené v týchto stromoch sú rovnaké (t.j. oba výpočty vedú k tomu istému výsledku). Ak sú rovnaké, výsledok akceptujú. Programátori z PAZ1b vedia, že pre jednu množinu hodnôt existuje veľa rôznych binárnych vyhľadávacích stromov, ktoré ju môžu uchovávať. Takže nestačí porovnať, či majú stromy rovnakú štruktúru. Na strane druhej si z midtermu spomínajú, že v čase $O(n)$ ide zistiť, či dva binárne vyhľadávacie stromy uchovávajú rovnakú množinu hodnôt.

**Úloha:** Uvažujme triedu BVS zo 4. prednášky predmetu PAZ1b. Do tejto triedy doplňte metódu:

```
public boolean rovnakeHodnoty(BVS vzor)
```

Metóda nech vráti **true** práve vtedy, keď strom obsahuje rovnakú množinu hodnôt ako strom referencovaný parametrom vzor. Časová zložitosť metódy musí byť $O(n)$, kde $n$ je počet hodnôt uložených v strome.

**Hodnotenie**:

- 6 bodov za riešenie s pamäťovou zložitosťou $O(n)$
- 18 bodov za riešenie s pamäťovou zložitosťou $O(1)$