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.

Školský výlet (15 bodov, backtracking)

Chatová osada PAZ-Village je častým cieľom školských výletov. V osade sú rôzne chatky – väčšie i menšie (2-lôžkové, 3-lôžkové, ..., 10-lôžkové, ...). Vždy, keď dôjde výlet, správca osady stojí pred neľahkou úlohou. K dispozícii má zoznam voľných chatiek (o každej samozrejme vie, koľko je v nej lôžok) a vie, koľko chlapcov a koľko dievčat prišlo na výlet v rámci školského výletu. Jeho cieľom je umiestniť výletníkov do chatiek tak, aby:

- žiadna chatka nebola „zmiešaná“ (chatka je obsadená buď celá chlapcami alebo celá dievčatami) – podmienka pedagogického dozoru,
- v každej chatke, ktorá bude výletníkom pridelená, boli všetky lôžka obsadené (cieľom je maximálne vyťažiť chatky).

Vytvorte program, ktorý načíta údaje o voľných chatkách (o každej voľnej chatke načíta počet jej lôžok), počet chlapcov a počet dievčat na školskom výlete. Spôsob a prípadne formát zadania vstupu nie sú nijako predpísané (vlastná voľba). Program následne vypíše také rozdelenie/pridelenie chatiek výletníkom (ak existuje), ktoré spĺňa vyššie uvedené podmienky. Ak existuje viacero vyhovujúcich rozdelení chatiek, nájdite to, ktoré obsadzuje čo najmenší počet chatiek.

Príklad:
Voľné chatky: *3 4 2 2 5 3 2 2 6*
Počet chlapcov: *9* Počet dievčat: *7*
Riešenie:
***3** (D) **4** (D) **2** (V) **2** (V) **5** (V) **3** (CH) **2** (V) **2** (V) **6** (CH)*
Počet obsadených chatiek: 4
Legenda: V – voľná chatka, D – chatka pre dievčatá, CH – chatka pre chlapcov

Vyhľadávanie s preklepmi (10 bodov, stringológia)

V Jave objekty triedy String obsahujú metódu **indexOf**, ktorá vráti prvý výskyt zadaného podreťazca v reťazci (CTRL+F funkcionalita). V praxi sa však stáva, že používateľ spraví preklep a tak reťazec, ktorý chce nájsť (*ihla*), zadá s preklepom alebo je preklep v samotnom texte (*seno*), v ktorom sa reťazec hľadá. Preto je užitočné mať metódu, ktorá bude vyhľadávať pozíciu podreťazca v reťazci, avšak bude odolná aj voči preklepom. Vytvorte metódu:

```
public static int vyhladajSPreklepom(String seno,
                        String ihla, int maxPreklep)
```

Táto metóda vráti takú pozíciu v reťazci **seno**, že reťazec **ihla** sa od podreťazca začínajúceho na tejto pozícii líši v čo najmenej znakoch. V prípade, že viac pozícii spĺňa túto podmienku, vráťte prvý výskyt. Ak sa podreťazec na nájdenej pozícii líši od reťazca **ihla** o viac než **maxPreklep** znakov, metóda nech vráti hodnotu -1.
Príklad:
seno = VITAJTE NA ZAVERECNOM **TESR**E Z PAZ1B *ihla* = TEST *najlepšia zhoda*: 1 preklep

Najdlhší podreťazec, ktorý je palindróm (15 bodov, dynamické programovanie)

Uvažujme reťazec $A = a_1a_2a_3a_4 \cdots a_n$ dĺžky n. Označme $A_{i..j}$ podreťazec reťazca A, taký že $A_{i..j} = a_ia_{i+1} \cdots a_j$. Podreťazec $A_{i..j}$ nazývame palindrómom, keď platí, že $a_{i+h} = a_{j-h}$ pre všetky $0 \leq h \leq j - i$. Inak povedané, palidróm je taký reťazec, ktorý sa číta rovnako odpredu aj odzadu. Príkladom palindrómu môžu byť reťazce aa, $abcba$, c alebo $ababcbaba$. Vytvorte program, ktorý pre zadaný reťazec $A = a_1a_2a_3a_4 \cdots a_n$ nájde dĺžku najdlhšieho jeho podreťazca, ktorý je palindróm.

Návod: Pre každé i a j také, že $1 \leq i \leq j \leq n$ si definujme výraz $P[i,j]$, ktorý bude nadobúdať hodnotu **true** práve vtedy, keď podreťazec $A_{i..j}$ je palindróm, resp. hodnotu false ak podreťazec $A_{i..j}$ palindróm nie je. Pre každý podreťazec dĺžky jedna zrejme triviálne platí, že to je palindróm, takže $P[i,i]$ = true pre všetky i. Ak si zoberieme ľubovoľný podreťazec dĺžky 2, tak zrejme platí, že $P[i,i+1]$ = **true** práve vtedy, keď $a_i = a_{i+1}$. Ak si potom zoberieme ľubovoľný podreťazec dĺžky aspoň 3 (teda podreťazec $A_{i..j}$, kde $j - i + 1 \geq 3$), tak pre neho zrejme platí, že to bude palindróm práva vtedy, keď jeho podreťazec $A_{i+1\,..\,j-1}$ palindróm je a taktiež platí, že prvé a posledné písmeno sú zhodné ($a_i = a_j$), takže $P[i,j]$ = true práve vtedy, keď $P[i+1, j-1]$ = true a súčasne $a_i = a_j$.

WSN (20 bodov, grafové algoritmy)

Bezdrôtová senzorová sieť (WSN) je komunikačná sieť tvorená veľkým počtom jednoduchých senzorov (merajú napríklad teplotu, detegujú pohyb, atď.) vybavených zariadeniami na rádiovú komunikáciu - transceivermi. Každý transceiver má svoj vysielací dosah r, ktorý hovorí, že signál vyslaný týmto transceiverom môžu prijať všetky iné transceivery vzdialené najviac r (uvažujeme Euklidovskú vzdialenosť) od vysielajúceho transceivera. Okrem senzorov sa v sieti nachádza jedno alebo viac tzv. base-station zariadení, ktoré zbierajú údaje zo senzorov a cez ďalšie pripojenie (napr. satelitné, GSM, ...) ich preposielajú do centrály. Ak v dosahu senzora (resp. transceivera na tomto senzore) nie je žiadna base-station, údaje zo senzora sa musia sprostredkovane preposielať cez ďalšie senzory (jeden alebo viac) v sieti. Každé vysielanie trvá jednu časovú jednotku (napr. 10 ms).

Textový súbor obsahuje popis bezdrôtovovej senzorovej siete. V prvom riadku sa nachádza počet zariadení tvoriacich bezdrôtovú senzorovú sieť - n. V ďalších n riadkoch sa v každom riadku nachádzajú medzerami oddelené informácie o jednom zariadení. Prvé 2 reálne čísla určujú x-ovú a y-ovú súradnicu pozície zariadenia. Ak je zariadenie senzor, za súradnicami sa nachádza kladné reálne číslo r, ktoré vyjadruje vysielací dosah transceivera na tomto senzore. Ak ide o zariadenie typu base-station, za súradnicami sa nachádza písmeno B označujúce, že ide o base-station.

Príklad vstupu:

```
5
5 4 6
1 2 9
15 4 B
7 0 4
10 1 9
```

V sieti je jedno base-station zariadenie a 4 senzorové zariadenia. Keďže vysielací dosah zariadenia na súradniciach [5, 4] je 6, tento senzor dokáže poslať správu (s nameranými hodnotami) do senzoru na súradniciach [1, 2], no nedokáže poslať správu priamo do base-station na súradniciach [15, 4].

Vytvorte program, ktorý načíta opis bezdrôtovej senzorovej siete a vráti, aký najmenší čas je potrebný na to, aby sa nameraná hodnota z každého senzora doručila do niektorej base-station. Ak existuje senzor, z ktorého nie je možné namerané hodnoty doručiť do niektorej z base-station, program nech o tom informuje používateľa.

Vkladanie podzoznamov (6 bodov)

Uvažujme triedu *SpajanyZoznam* z prednášky o spájaných zoznamoch. Do triedy *SpajanyZoznam* pridajte metódu *vlozPodzoznam*, ktorá do spájaného zoznamu od zadanej pozície (indexu) vloží postupnosť hodnôt uloženú v poli *hodnoty*. Metóda nech pracuje v lineárnom čase vzhľadom k dĺžke zoznamu a poľa hodnoty.

```
public void vlozPodzoznam(int index, int[] hodnoty)
```

Príklad: Predpokladajme, že spájaný zoznam obsahuje hodnoty `[8, 5, 2, 5, 1, 4]`. Volanie metódy `vlozPodzoznam(3, [2, 8, 7])` spôsobí, že spájaný zoznam bude obsahovať hodnoty `[8, 5, 2, 2, 8, 7, 5, 1, 4]`.