



Závěrečný test praktická část

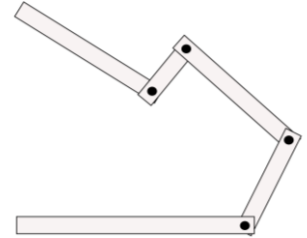


Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Doplňujúce zdrojové kódy sú na stránke predmetu PAZ1b. Funkčnosť každého riešenia musí byť preukázaná spustením na testovacom vstupe - nespustiteľné riešenia neumožňujú zisk príslušných bodov.

Skladací meter (12 bodov, backtracking)

Firma *PAZomer* sa rozhodla vyrábať netradičné skladacie metre. Na rozdiel od tých klasických sa tieto metre neskladajú z dielov rovnakej dĺžky, ale každý diel môže mať inú dĺžku. Obchod sa rozbehol a prišlo mnoho objednávok. Problém však vznikol pri balení. Keďže je žiaduce, aby balenie bolo čo najmenšie, vznikla otázka: na akú najmenšiu dĺžku je možné poskladať meter (meradlo)? Vytvorte program, ktorý vyrieši túto otázku pre zadanú postupnosť dĺžok jednotlivých dielov skladacieho metra a vráti poskladanie, v ktorom sa dosahuje táto najmenšia dĺžka. Vstup načítajte z textového súboru.



Príklad: Ak sa meter skladá z dielov s dĺžkami 15 10 12, tak najmenšie možné poskladanie má dĺžku 17.

Rada: Ak sa meter skladá z n dielov, tak obsahuje $n - 1$ kĺbov. Každé poskladanie metra je určené zalomením/nezalomením jednotlivých kĺbov.

Najdlhšia palindromická podpostupnosť (7+7 bodov, dynamické programovanie)

Hľadanie najdlhších podpostupností a palindrómové veci sú už také klasické úlohy. Prečo to ale nespojiť a neskúsiť nájsť najdlhšiu vybranú podpostupnosť zadaného reťazca, ktorá je palindrómom? Uvažujme reťazec *cacabakla*. Tento reťazec má mnoho podpostupností (napr. *ccaba*, *acakla*, ...). Nás ale zaujímajú palindromické podpostupnosti (napr. *aca*, *aaaa*, *aka*, ...). V reťazci *cacabakla* je však najdlhšia palindromická podpostupnosť reťazec *aabaa* (viď: *ca**cab**akla*).

Úloha: S využitím dynamického programovania naprogramujte metódu, ktorá pre zadané slovo nájde jeho najdlhšiu palindromickú podpostupnosť.

Návod: Uvažujme reťazec $r_1 r_2 r_3 \dots r_n$ dĺžky n . Označme si $P[i, k]$ dĺžku najdlhšej palindromickej podpostupnosti reťazca $r_i r_{i+1} r_{i+2} \dots r_k$. Triviálne $P[i, i] = 1$. Vo všeobecnosti platí:

$$P[i, k] = \begin{cases} \max\{P[i + 1, k], P[i, k - 1]\}, & r_i \neq r_k \\ P[i + 1, k - 1] + 2, & r_i = r_k \end{cases}$$

Neklesajúce zoznamy (6 bodov, spájané zoznamy)

Uvažujme triedu *SpajanyZoznam* z prednášky o spájaných zoznamoch. Do triedy *SpajanyZoznam* pridajte metódu *vlozSpravne*, ktorá do spájaného zoznamu vloží vkladajú hodnotu pred prvú položku zoznamu, ktorá je väčšia ako vkladaná hodnota, resp. na koniec zoznamu, ak takej položky niet. Výsledkom vkladania len cez *vlozSpravne* je zachovanie usporiadania prvkov zoznamu.

Príklad: Uvažujme spájaný zoznam: [4, 5, 10, 13, 19, 35]. Potom volaním metódy:

- `vlozSpravne(15)` vznikne zoznam [4, 5, 10, 13, 15, 19, 35]
- `vlozSpravne(2)` vznikne zoznam [2, 4, 5, 10, 13, 19, 35]
- `vlozSpravne(42)` vznikne zoznam [4, 5, 10, 13, 19, 35, 42]
- `vlozSpravne(19)` vznikne zoznam [4, 5, 10, 13, 19, 19, 35]

```
public void vlozSpravne(int cislo)
```

Efektívnosť implementácie: Nech n je dĺžka spájaného zoznamu. Časová zložitosť metódy `vlozSpravne` musí byť $O(n)$ a pamäťová $O(1)$.

PAZOceánia (16+7 bodov, grafové algoritmy)

PAZOceánia je zaujímavý kontinent plný ostrovov so skvelým potenciálom na rozvoj cestovného ruchu. Základným spôsobom dopravy je celkom prirodzene lodná doprava tvorená množstvom nepravidelných lodných spojení. Každé takéto spojenie spája nejaké dva ostrovy. Lode v PAZOceánii nie sú veľmi rýchle a vzdialenosti medzi ostrovmi sú veľké. Plavba tak trvá jeden alebo aj viacero dní. Lod' vypláva z prístavu v deň plavby vždy o 8:00. Keďže tých ostrovov je tak veľa, že im miestni obyvatelia nevedia vymyslieť mená,



namiesto mien si ich len očíslovali. PAZOceánia sa skladá z ostrovov $0, 1, 2, \dots, n - 1$. Rozvoj internetu zasiahol aj PAZOceániu a dnes každý obyvateľ PAZOceánie má mobil s internetovým pripojením. Jednoducho skvelé podmienky na informatizáciu. Prvá webová aplikácia, ktorá má pomôcť obyvateľom a návštevníkom PAZOceánie je www.pp.paz (plavebné poriadky). Používateľ len zadá odkiaľ a kam chce cestovať (plaviť sa) a portál mu nájde najkratšie plavebné spojenie zo zadaného štartovacieho ostrova do zadaného cieľového ostrova.

Zadanie: Vytvorte program, ktorý načíta zoznam nepravidelných lodných spojení. Program potom pre zadaný dátum (ako poradové číslo dňa v roku) a zadané dva ostrovy nájde také spojenie, ktorým sa môžeme čo najskôr po zadanom dátume dostať na cieľový ostrov. Poznamenajme, že aj čas čakania na lodné spojenie sa zarátava do času cesty.

Opis vstupu: Nepravidelné lodné spojenia sú opísané v textovom súbore. Prvý riadok súboru obsahuje počet ostrovov v PAZOceánii (n) a počet lodných spojení (m). Každé lodné spojenie je opísané v samostatnom riadku a skladá sa z medzerami oddelených čísel $S C D T$. Jednotlivé čísla znamenajú:

- S - číslo ostrova, z ktorého lod' vypláva,
- C - číslo ostrova, do ktorého lod' pláva,
- D - dátum ako poradové číslo dňa v roku, kedy lod' vypláva,
- T - trvanie plavby v dňoch.

Ak lod' vypláva v deň D , tak do cieľa dôjde v deň $D + T - 1$. V ďalšej plavbe (ak nie sme v cieľi) môžeme pokračovať až v deň $D + T$.

Rada: Samotná informácia, že som na ostrove, necharakterizuje stav cestovania. Konkrétne, ak som na ostrove č. 5, tak nemôžem pokračovať ľubovoľným lodným spojením ďalej. Môžem ísť len tými loďami, ktoré z ostrova č. 5 vyplávajú až po dni, kedy som na ostrov č. 5 prišiel. Stav cestovateľa preto lepšie charakterizuje dvojica (číslo ostrova, deň). Napríklad dvojica (5, 30) môže byť interpretovaná tak, že v poradí 30-ty deň roku som na ostrove č. 5.