



Záverečný test Zadanie



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Doplňujúce zdrojové kódy sú na stránke predmetu PAZ1b. Funkčnosť každého riešenia musí byť preukázaná spustením na testovacích vstupoch - nespustiteľné riešenia neumožňujú získanie príslušných bodov.

Cica Mica a Kocúr Jonatán

Petra má cicu Micu, ktorá každú noc behá niekde po vonku spolu s kocúrom Jonatánom. Nevráti sa skôr než ráno a cez deň spí ako bábo, celý deň spí na peci a to Petru vôbec neteší. Cez deň sa s ňou nechce hrať lebo musí spať. Preto sa Petra rozhodla, že skúsi zaujať Micu novými hračkami alebo aktivitami a ak sa to nepodarí tak sa sama niekam vyberie na výlet a nechá Cicu behať cele noci za kocúrom Jonatánom.

Hračky pre Micu (18 bodov, backtracking)

Petra sa rozhodla kúpiť pre Micu rôzne hračky aby sa s ňou viac hrala. Ako pozerá po e-shopoch, tak nie každý má všetko v ponuke a ešte aj rôzne e-shopy pristupujú rôzne ku poštovnému. Občas sa dokonca oplatí zobrať drahšie produkty len aby bolo poštovné zadarmo. Preto si Petra spravila zoznam, kde má napísané pri každom predajcovi cenu poštovného, cenu nákupu od ktorej je poštovné zadarmo a zoznam cien hračiek o ktoré má záujem napr.:



Predajca	Poštovné	Poštovné zadarmo od	Pierko na paličke	Mačací hrad	Škrabadlo	Pelech
macky.sk	15	50	25	X	10	25
C&M cats	20	45	X	25	20	22
Predajca 3	30	100	30	30	30	30

Petra sa zamyslela a hneď prišla na to, že pomocou backtracku vie vypočítať, ako zaplatiť za všetky hračky s dopravou čo najmenej.

Pozn. Petra kupuje z každého tovaru práve jeden kus.

Úlohy: Implementujte program, ktorý načíta predajcov a informácie o každom z nich - poštovné, sumu od ktorej je poštovné zadarmo a predajné ceny hračiek. Formát vstupu si zvolte podľa vlastného uváženia.

(10b) Program vypíše najmenšiu možnú cenu za nákup všetkých hračiek a poštovné spojené s ich prepravou.

(+3b) Za výpis, ktoré hračky treba kúpiť od ktorého predajcu.

(+5b) Podľa efektívnosti. Včasné vyhodnotenie prípustnosti riešenia prípadne generovanie, ktoré bude mať menej ako 2^n možností.

Pozn.: Pre jednoduchosť stačí ak budete predajcov aj hračky číslovať napr. 0, 1, 2, ...

Oddych pre cicu (14 bodov, grafové algoritmy)

Keď sa cica Mica vráti ráno, tak je unavená a chce si ľahnúť niekde v miestnosti ale nie hocikde. Kto videl mačky oddychovať vie, že si vyberajú zvláštne miesta. Občas vysoko inokedy sú schované za nábytkom, alebo len prejdú po miestnosti na slnečnú stranu. Unavená Mica, chce prejsť na svoje miesto tak, aby minula čo najmenej energie. Je lepšie vyskočiť najprv na stoličku a potom na stôl ako spraviť jeden obrovský skok rovno na stôl.



Úlohy:

(9b) Implementujte program, ktorý načíta mapu miestnosti, súradnicu, kde je Mica teraz a kde chce ísť spať. Metóda vypíše cenu energeticky najmenej náročnej cesty. Mapa určuje výšku jednotlivých predmetov a podlahy. Mica sa pohybuje po mape len v smeroch $\leftarrow \uparrow \rightarrow \downarrow$. Energia na prechod z políčka na jeho suseda je $1 + (\text{rozdiel výšok})^2$. Ak ide Mica z políčka ktoré je vyššie na políčko ktoré je nižšie (zoskočenie), tak je energia prechodu práve 1.

Jedna vzorová mapa je zobrazená dole. Vo vzorovej mape sú riadky označené písmenami a stĺpce číslami. Formu reprezentácie mapy, súradníc a vstupných dát vo vašej implementácii si môžete zvoliť podľa svojich preferencií.

	0	1	2	3
A	1	1	1	1
B	1	2	1	1
C	1	3	5	1
D	1	1	1	1

Napr. Z C3 na C2 je energia $1 + (5-1)^2 = 17$. Ale ak pôjde Mica z C3 na C2 cestou C3 B3 B2 B1 C1 C2 tak na to potrebuje iba energiu $1 + 0^2 + 1 + 0^2 + 1 + 1^2 + 1 + 1^2 + 1 + 2^2$, čo je spolu iba 11.

(+5b) Algoritmus vypíše cestu ktorá je energeticky pre cicu najmenej náročná.

Mica zhadzovačka (13 bodov, dynamické programovanie)

Keď sa cica Mica zobudila tak si všimla, že u nich prebieha oslava a všetci sa bavia. Tak si pomyslela „Všetci sú takí šťastní a spokojní.“, usmiala sa, „Ved' ja to dokážem zmeniť.“ Preto sa rozhodla, že prebehne po stole aby zhodila čo najviac pohárov, tanierov vidličiek a ďalších vecí. Pozrela na stôl a nevedela zrazu ako bežať ved', mohla prebehnúť vpravo, vľavo, stredom alebo niekoľkokrát zmeniť kade beží.



Úlohy: (7b) Navrhните algoritmus, ktorý načíta počet vecí v jednotlivých častiach stola. Mica beží z vrchu smerom dole, pričom vždy môže ostať bežať v tej časti kde je ↓ alebo môže ísť smerom doprava dole ↘ alebo doľava dole ↙. Začať môže na ľubovoľnej časti stola. Metóda vypíše koľko najviac vecí môže zhodiť. Riešenie bude v polynomiálnom čase vzhľadom k veľkosti vstupu.

Príklad

5	1	1
3	2	1
2	5	8
1	2	4
9	5	4

5	1	1
3	2	1
2	5	8
1	2	4
9	5	4

Na červenej ceste na obrázku vľavo Mica na ceste zhora dole zhodí 16 vecí, čo nie je najviac vecí koľko môže zhodiť. Ak pôjde po červenej ceste ako na obrázku vpravo, tak zhodí najviac vecí, až 26.

(+6b) Metóda vypíše cestu kedy Mica zhodí najviac predmetov, formát cesty si môžete zvoliť podľa seba.

Alicin playlist (5 bodov, spájané zoznamy)

Alica si počas cesty na diaľnici ku svojej športovej jazde rada púšťa pesničky (riadne vypaľovačky) zo svojich rôznych playlistov. Aby v dobrej nálade nezabudla odbočiť z diaľnice prišla na to, že posledná pesnička musí dohrať pred zjazdom z diaľnice. Keďže vie ako dlho trvá jej jazda po diaľnici, chce to využiť. Poradie v playlistoch si nechce meniť ale chce z nich iba z konca odrezat' pesničky tak aby nebol playlist dlhší ako jej cesta.



Zdroj:
Hillard art museum
Lafayette, LA, USA

Uvažujme triedu *SpajanyZoznam* z prednášky o spájaných zoznamoch. Do triedy *SpajanyZoznam* pridajte metódu *orezPodDlzkou* s parametrom *dlzka*, ktorá odreže zo spájaného zoznamu z konca čísla tak aby súčet čísel, ktoré ostanú v zozname bol menší ako hodnota zadaná parametrom *dlzka*.

Príklad: Uvažujme spájaný zoznam: [8, 5, 5, 10, 2, 6, 14]. Potom volaním metódy:

- *orezPodDlzkou*(14) sa zmení zoznam na [8, 5],
- *orezPodDlzkou*(2) sa zmení zoznam na [],
- *orezPodDlzkou*(13) sa zmení zoznam na [8],
- *orezPodDlzkou*(20) sa zmení zoznam na [8, 5, 5],
- *orezPodDlzkou*(142) sa zmení zoznam na [8, 5, 5, 10, 2, 6, 14].

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

Rozbité stromy (12 bodov, stromy)

Keď sa v kreatívnom laboratóriu učili o stromoch spolu Terky, Kevin, Alica, Sophia, Andrii, Feri a kto vie kto ešte ďalší, tak im na stôl vyskočila Mica. Po skoku si všimli, že ich tri reprezentácie toho istého stromu sú rozbité, tak ich rýchlo uložili naspäť a škrabali si hlavu, či teraz tieto tri reprezentácie toho istého stromu naozaj reprezentujú ten istý strom alebo nie. Zrazu všetci naraz zakričali na celý kreatív „Ved' si to naprogramujme.“

Úloha: Implementujte program, ktorý načíta stromy v troch rôznych reprezentáciách a overí, či zodpovedajú identickému stromu. Navyše vieme, že každá hodnota v strome sa nachádza najviac raz.

Rôzne reprezentácie:

1. Referencia na koreň stromu, ktorý je reprezentovaný triedou *Osoba* zo 4. prednášky.
2. Strom je reprezentovaný dvojicou polí, kde hodnoty na rovnakých indexoch spolu súvisia. Každý index reprezentuje jednu osobu. Prvé pole obsahuje meno osoby a druhé pole obsahuje meno rodiča danej osoby alebo null, ak je koreňom stromu (predpokladáme, že každá osoba má unikátne meno).
3. Pole ret'azcov a matica susednosti orientovaného grafu, pričom orientovaný graf uložený v matici určuje potomkov a rodičov. Hrana ide od rodiča k potomkovi. Hodnota v poli na indexe i vyjadruje meno i -tej osoby.

Bodovanie: Za implementáciu (metódy?) porovnávania každej dvojice reprezentácií 1.-2., 1.-3., 2.-3. sú 4 body.