



Závèrečný 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.

Minulý týždeň sa Slovenskom rýchlosťou vyše 40 km/hod prehnal pelotón pretekárov cyklistickej súťaže *Okolo Slovenska 2014*. Závèrečnú etapu dlhú 178.6 km v letných horúčavách prešli za 4 hodiny a 10 minút. Koľko úloh za tento čas možno vyriešiť na teste z PAZ1b?



Vrchársky okruh (12+6 bodov, backtracking)

Organizátori pretekov sa rozhodli pripraviť jednu vrchársku okružnú etapu. Vrchárska etapa znamená, že by tam malo byť dosť veľa nastúpaných metrov. Okružná etapa znamená, že štart a cieľ etapy je na tom istom mieste (každým úsekom cesty a každou križovatkou sa prechádza len raz). Preteky sa nemôžu robiť na akýchkoľvek cestách a tak organizátori vytipovali niektoré úseky ciest, ktoré sú vhodné. Z vhodných úsekov zostavili jednoduchý orientovaný ohodnotený graf. Každý úsek predstavuje hranu grafu. Konce úsekov (križovatky) sú vrcholy grafu. Ohodnotenie hrany (úseku) zodpovedá počtu nastúpaných metrov na tomto úseku (ak ide úsek len z kopca alebo je to rovina, nastúpané metre úseku sú 0). Počet nastúpaných metrov celej etapy je súčet nastúpaných metrov na jednotlivých úsekoch etapy.

Zadanie: V textovom súbore (formát podľa vlastného uváženia) je popísaný orientovaný ohodnotený graf. Nájdite taký okruh (pozor, nemusí prechádzať cez všetky križovatky, resp. začiatky a konce úsekov = vrcholy grafu), ktorý má najväčší možný celkový počet nastúpaných metrov.

Bodovanie: 6 b je za riešenie, v ktorom sa negenerujú všetky variácie s opakovaním vrcholov.

Prechod mestom (14+7 bodov, grafové algoritmy)

Prechod pelotónu mestom je z hľadiska organizácie dopravy komplikovanejší. Aj preto organizátori majú záujem prejsť väčšími mestami čo najrýchlejšie, t.j. čo najkratšou cestou. No najkratších ciest prechádzajúcich mestom môže byť viacej. V takomto prípade chcú organizátori vybrať takú cestu, na ktorej je najmenej križovatiek (z grafového pohľadu sa cesta skladá z najmenšieho počtu hrán). Totiž na každej križoviatke musia byť policajti, ktorí by blokovali vjazd áut do križovatky. Čiže každá križoviatka znamená jednu policajnú hliadku, resp. na nejaký čas zablokovanú križoviatku.

Zadanie: V textovom súbore (formát podľa vlastného uváženia) je uložený ohodnotený graf popisujúci ulice mesta. Hrany zodpovedajú uliciam (alebo ich častiam), vrcholy križovatkám. Nájdite najlacnejšiu (najkratšiu) cestu medzi dvoma zadanými (napr. z konzoly alebo zo súboru) križovatkami (vrcholmi grafu), ktorá má navyše tú vlastnosť, že spomedzi najlacnejších (najkratších) ciest sa skladá z najmenšieho počtu hrán.

Rada: Ako ohodnotenie cesty neuvažujte len jedno číslo d vyjadrujúce súčet ohodnotení hrán cesty, ale usporiadanú dvojicu (d, h) , kde d je súčet ohodnotení hrán a h je počet hrán na tejto ceste. Teda aj odhady dĺžky najkratších ciest sú takéto dvojice, ktoré porovnávame lexikograficky. Pritom platí, že $(d_1, h_1) < (d_2, h_2) \Leftrightarrow d_1 < d_2 \vee (d_1 = d_2 \wedge h_1 < h_2)$.

Bodovanie: 14 b za dĺžku najkratšej cesty spolu s počtom hrán na nej, 7 b za jej výpis.

Sponzoring (13 bodov, dynamické programovanie)

Cyklistické tímy by bez sponzorov len ťažko prežili. Sponzorské zanietenie jednej grófskej rodiny ju takmer priviedlo na mizinu. V dôsledku finančných problémov začali rozpredávať víno. Ich príbeh nájdete v priložených slajdoch (autorom je Ján Jerguš [Facebook, USA], ktorý v tom čase ako prvák kolegom vysvetľoval riešenie úlohy).

Zadanie: Upravte riešenie úlohy tak, aby ste vypísali aj poradie, v akom sa jednotlivé vína majú predávať.

Cykli(sti)cké zoznamy (max. 8 bodov, spájané zoznamy)

Uvažujme triedu `SpajanyZoznam` z prednášky o spájaných zoznamoch. Do triedy `SpajanyZoznam` pridajte metódu `cyklickyShift`. Táto metóda nech `cyklicky` posunie všetky UZLY v spájanom zozname vpravo o zadaný počet pozícií (parameter `posun`). Ak je hodnota `posun` záporná, hodnoty v zozname nech sa posúvajú vľavo.

```
public void cyklickyShift(int posun)
```

Príklady: Uvažujme zoznam s hodnotami [3, 5, 1, 8, 10, 4, 9].

- `cyklickyShift(0)`: [3, 5, 1, 8, 10, 4, 9]
- `cyklickyShift(1)`: [9, 3, 5, 1, 8, 10, 4]
- `cyklickyShift(2)`: [4, 9, 3, 5, 1, 8, 10]
- `cyklickyShift(3)`: [10, 4, 9, 3, 5, 1, 8]
- `cyklickyShift(-1)`: [5, 1, 8, 10, 4, 9, 3]
- `cyklickyShift(-2)`: [1, 8, 10, 4, 9, 3, 5]
- `cyklickyShift(7)`: [3, 5, 1, 8, 10, 4, 9]

Požiadavky na implementáciu:

- Preusporiadať sa majú uzly spájaného zoznamu, hodnoty uzlov sa nekopírujú.
- Pamäťová zložitosť: $O(1)$
- Časová zložitosť pri zozname dĺžky n a posune o k :
 - $O(k \cdot n)$ - 5 bodov
 - $O(n)$ - 8 bodov