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.

CTRL+A/CTRL+C, CTRL+V (9+7 bodov, dynamické programovanie)

Poznámkový blok (notepad) na začiatku obsahuje len písmeno A. Kopírovacia schránka (clipboard) je prázdna. Môžete vykonať 2 operácie:

- COPY ALL – skopíruje všetok obsah poznámkového bloku do schránky,
- PASTE – prilepí obsah schránky na koniec obsahu poznámkového bloku.

Pre zadané číslo n $(n > 0)$ nájdite minimálny počet operácii, ktoré je treba vykonať, aby poznámkový blok obsahoval práve n písmen A. Poznamenajme, že takýto stav je možné vždy dosiahnuť postupnosťou operácii: COPY ALL, $(n-1)$ x PASTE - teda celkom n operáciami. Ale dá sa dosiahnuť aj menej...

Rada: Označme si $R[i,j]$ minimálny počet operácii, ktoré musíme spraviť, aby sme v poznámkovom bloku mali i písmen A a v kopírovacej schránke j písmen A. Východisková situácia zodpovedá $R[1,0]$ a teda $R[1,0] = 0$. Z tohto stavu sa operáciou COPY ALL vieme dostať do stavu $[1,1]$, t.j. $R[1,1] = 1$. Niektoré stavy charakterizované hodnotami i a j nie je možné dosiahnuť - napr. $[5,0]$, či $[1,100]$. Dokonca platí, že ani stavy $[i,j]$, kde $j > i$ nemožno dosiahnuť (prečo?). Podobne ako v úlohe o vyplatení dlhopisov môžeme použiť špeciálnu hodnotu -1 indikujúcu, že daný stav nemôže nastať. Teda $R[i,j] = -1$, ak stav $[i,j]$ nemožno dosiahnuť z východiskového stavu. Zamyslite sa: Ak máme v poznámkovom bloku i písmen a v schránke j písmen, z akých stavov sa vieme dosať do tohto stavu? Príklad: Ak mám stav $[8,3]$, potom jediná možnosť ako ho dosiahnuť, je spraviť PASTE v stave $[5,3]$. Stav $[5,5]$ vieme dosiahnuť operáciou COPY ALL z niektorého zo stavov $[5,1], [5,2], [5,3], [5,4]$.

Hodnotenie: 9 bodov za nájdenie minimálneho počtu operácii v polynomiálnom čase, 7 bodov za vrátenie postupnosti operácii, kedy sa tento minimálny počet operácii dosahuje.

Návraty k midtermu (5 bodov)

Vieme, že platí toto tvrdenie: *„V čase $O(n.logn)$ môžeme zistiť, ktorá hodnota sa v n-prvkovom poli vyskytuje najväčší počet krát.“*

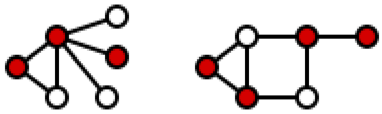
Naprogramujte metódu, ktorá v uvedenom (aj priemernom) čase nájde tú hodnotu v poli, ktorá sa v ňom vyskytuje najväčší počet krát. Ak je takých hodnôt viac, metóda nech vráti ľubovoľnú z nich. Metóda nesmie modifikovať referencované pole p. Akceptovaná pamäťová zložitosť je $O(n)$.

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

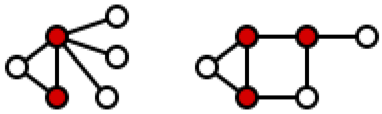
Najlacnejšie vrcholové pokrytie (14 bodov, backtracking)

Klasickým grafovo-algoritmickým problémom je tzv. problém najlacnejšieho vrcholového pokrytia jednoduchého neorientovaného grafu. O čo ide?

Vrcholovým pokrytím grafu $G = (V, E)$ nazývame každú podmnožinu vrcholov grafu $C \subseteq V$ takú, že pre každú hranu $e = (u, v)$ grafu G platí $u \in C \vee v \in C$. Teda <u>aspoň jeden</u> koniec každej hrany grafu musí patriť do množiny C. Príklady vrcholových pokrytí:



Triviálnym vrcholovým pokrytím je napríklad množina všetkých vrcholov grafu. Graf môže mať veľa vrcholových pokrytí. Preto má zmysel hľadať akýmsi spôsobom „najlepšie“ vrcholové pokrytie. Ak by sme uvažovali ako mieru „kvality“ počet vrcholov vo vrcholovom pokrytí, hľadali by sme minimálne vrcholové pokrytie:



V praxi ale výber vrcholu do vrcholového pokrytia môže mať rôzne ceny. To vieme zachytiť tak, že každému vrcholu priradíme cenu za jeho vybranie do vrcholového pokrytia. Rôzne vrcholy môžu mať rôzne ceny. Vytvorte program, ktorý pre zadaný graf nájde jeho najlacnejšie vrcholové pokrytie, t.j. pokrytie, v ktorom súčet cien vybraných vrcholov je najmenší možný.

Formát vstupu:
V prvom riadku sa nachádzajú 2 čísla: počet vrcholov N a počet hrán M. Vrcholy grafu budeme indexovať číslami 0..N-1.
V druhom riadku sa nachádza N reálnych čísel určujúci pre každý vrchol (v poradí od vrcholu 0) cenu jeho vybrania do vrcholového pokrytia.
V každom z ďalších M riadkov sa nachádza informácia o jednej hrane grafu. Táto informácia sa skladá z 2 čísel tvoriacich indexy vrcholov, ktoré táto hrana spája.

Príklad:
```
5 3
3 100 2 30 9
0 2
3 1
1 4
```

Rada: Aj keď je to úloha o grafoch, grafové algoritmy tu netreba použiť. Odporúčaný spôsob uloženia grafu je prostredníctvom zoznamu hrán.

Bipartitné grafy (12 bodov, grafy)

Zaujímavou skupinou grafov (aj z pohľadu praxe) sú takzvané bipartitné grafy. Aké to sú grafy? Graf G nazveme bipartitným, ak množinu jeho vrcholov $V(G)$ môžeme rozdeliť na dve disjunktné podmnožiny X a Y, t.j. $X \cap Y = \emptyset, X \cup Y = V(G)$, s takou vlastnosťou, že pre každú hranu grafu platí, že jeden jej koniec je v množine X a druhý v množine Y.

Úloha: Vytvorte program, ktorý pre zadaný neorientovaný (potenciálne aj nesúvislý) graf overí, či je tento graf bipartitný. Graf načítajte z textového súboru, formát si zvoľte podľa vlastného uváženia. Očakáva sa riešenie v polynomiálnom čase od veľkosti grafu.

Rada: Nech G je súvislý bipartitný graf. Ak si zoberieme ľubovoľný vrchol $s \in X$, tak jeho susedia (ak nejakých má) sú z množiny Y. Toto pozorovanie ide rozšíriť. Ak si zoberieme ľubovoľný vrchol $s \in X$, potom vrcholy, ktorých vzdialenosť od s je párna, sú v množine X a vrcholy, ktorých vzdialenosť od s je nepárna, sú v množine Y. Čo ale s hocijakým súvislým grafom? Nuž ak by takýto graf mal byť bipartitný, potom po rozdelení vrcholov do množín X a Y podľa vzdialenosti od zvoleného vrcholu by malo platiť, že v grafe niet hrany, ktorá by mala oba konce buď v X alebo Y.

Deň D (4+7 bodov)

Súostrovie PAZ Oceánia sa skladá z n ostrovov číslovaných $0 \dots n-1$. PAZ Oceánia získala grant na vybudovanie pravidelných obojsmerných lodných spojení medzi ostrovmi. Už je známy aj harmonogram, v akom sa jednotlivé spojenia budú postupne otvárať. Každý deň je v pláne otvoriť práve jedno nové (obojsmerné) lodné spojenie. Obyvatelia PAZ Oceánie sa už nevedia dočkať dňa, kedy sa bude dať cestovať medzi ľubovoľnými dvoma ostrovmi súostrovia.

Zadanie: V textovom súbore je zadané číslo n - počet ostrovov PAZ Oceánie. V každom z ďalších riadkov sa nachádza dvojica medzerou oddelených čísel x a y, ktoré označujú vytvorené obojsmerné lodné spojenie medzi ostrovmi s číslami x a y. Spojenia v súbore sú uložené chronologicky, t.j. podľa času, kedy sa uvedú do prevádzky. Vytvorte program, ktorý vráti, v koľký deň podľa harmonogramu sa bude dať po prvý krát cestovať medzi ľubovoľnými dvoma ostrovmi súostrovia (potenciálne aj s prestupmi).

Hodnotenie: 4b za riešenie v polynomiálnom čase + 7b za riešenie v čase $O(n^2)$.

Rada: Inšpirujte sa grafovými algoritmami, ktoré sú založené na postupnom pridávaní hrán do grafu a testovaní súvislosti.