



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.

PAZostav-sk

Viazané živnosti (8 + 6 bodov, backtracking)

Spoločnosť PAZostav-sk, ktorá sa venuje stavebnej činnosti a súvisiacim dokončovacím prácam, sa rozhodla, že v najbližšej dobe sa zapojí do štátnych zákaziek. Aby spoločnosť mohla mať vo svojom portfóliu vykonávanie niektorých činností, napríklad vykonávanie montáže plynových zariadení, musí uzavrieť zmluvu s odborníkom s danou viazanou živnosťou. Získanie viazanej živnosti je podmienené vzdelaním, oprávnením, praxou, osvedčením a/alebo absolvovaním kurzu, preto získanie týchto živností je odborne a finančne náročné. Aby PAZostav-sk mohla deklarovat' odborné vykonávanie činností, a tým sa mohla prihlásiť do výberového konania, musí mať podpísanú zmluvu so živnostníkom v danej oblasti. Uzavrieť dlhodobú dohodu so živnostníkom s viacerými viazanými živnosťami vôbec nie je lacná záležitosť, preto je potrebné rozhodnúť, s ktorými živnostníkmi je potrebné uzavrieť dohody, aby firma mala odborníka s každou relevantnou živnosťou a platila im čo najmenej. Vytvorte program, ktorý pre zadaný vstup nájde takých živnostníkov, ktorí spoločne pokrývajú všetky živnosti a náklady na ich zamestnanie sú najmenšie.

Formát vstupu: Prvý riadok obsahuje počet živnostníkov n a počet živností m . Každý ďalší riadok vstupu obsahuje celé číslo vyjadrujúce mesačné náklady a nasleduje m hodnôt **true/false**, ktoré vyjadrujú, či živnostník na danú činnosť má oprávnenie alebo nie.

Príklad:

```
4 5
3000 true true true false true
1700 false false true true true
1200 true false false false true
1500 false true false true true
```

Úlohy:

(8b) Implementujte program, ktorý načíta textový súbor a nájde takú voľbu živnostníkov, aby dokopy pokrývali všetky živnosti a ich mesačné náklady boli najmenšie.

(+6b) Podľa efektívnosti. Algoritmus je tým efektívnejší, či viac eliminujete výpočty, ktoré nevedú k prípustnému riešeniu.

Nový stavebný sklad (8 + 9 bodov, grafové algoritmy)

Keďže firme PAZostav-sk sa podarilo vyhrať mnoho výberových konaní, chce postaviť nový sklad stavebného materiálu a techniky. PAZostav-sk má mapu obcí (vrcholov), ciest medzi nimi (ohodnotené hrany) a aktuálne existujúcich skladov (vybrané vrcholy). Cieľom PAZostav-sk je vybrať miesto, kde bude stáť nový sklad tak, aby súčet časov dojazdov do všetkých obcí bol čo najmenší.

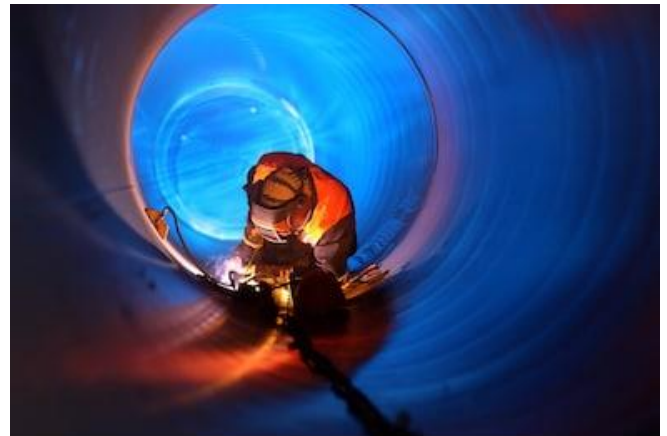


Úlohy: (8b) Vytvorte program, ktorý pre graf a zadané vrcholy reprezentujúce existujúce sklady vypočíta vzdialenosť každého vrcholu od najbližšieho skladu.

(+9b) Vytvorte program, ktorý pre zadaný graf a zadané vrcholy reprezentujúce existujúce sklady vráti, kde má byť postavený nový sklad aby súčet časov dojazdov do všetkých obcí bol čo najmenší.

Potrúbné siete (4 + 9 bodov, dynamické programovanie)

Stavební analytici spoločnosti PAZostav-sk si uvedomili, že potrubia sú často využívaným prvkom. Potrubia sa bežne využívajú na vedenie plynu, teplej a studenej vody, odpadu v kanalizácii a máme aj vzduchotechnické potrubia, alebo potrubnú poštu. Spoločné pre všetky druhy potrubí je, že čas spojenia dvoch potrubí (napr. zvárania) je omnoho menší ako čas rezania a spájania potrubí. Preto analytici potrebujú navrhnúť program, ktorý by zabezpečil návrh potrubí rôznej dĺžky bez potreby rezania. Program na vstupe dostane n typov rúr s dĺžkami $d[0], d[1], \dots, d[n-1]$ a dĺžku L , ktorá vyjadruje aké dlhé potrubie potrebujeme navrhnúť. Rozrastajúcej firme sa darí, preto má sklady dobre zásobené všetkými druhmi a prevedeniami potrubnej techniky, to znamená, že môžete predpokladať, že rúru s dĺžkou $d[i]$ môžeme použiť ľubovoľný počet krát.



shutterstock.com • 178021226

Úlohy:

(4b) Implementujte program, ktorý bude mať vstupné parametre $d[0], d[1], \dots, d[n-1]$ vyjadrujúce dostupné dĺžky rúr a parameter L vyjadrujúci dĺžku skladaného potrubia. Program nech vráti hodnotu **true/false**, či je možné z rúr vyskladať potrubie presne.

(+9b) Upravte program tak, aby vrátil koľko rúr akej dĺžky je potrebných na vyskladanie potrubia.

Diaľnice, tunely a obchvaty (7 + 7 bodov, grafové algoritmy?)

V našom krásnom kraji diaľnicu už pred rokmi stavať začali. Každý vie, aký to bude mať na našich ľudí dopad, všetkým sa bude žiť lepšie, zápchy v našom meste budú trochu redšie. Už nám závidia aj v Sabinove diaľničné spojenie nové. Aby nám nezávideli diaľničné spojenie v Sabinove ani žiadnom inom meste, preto je navrhnutý diaľničný stavebný plán. V pláne sú mesta očíslované 0, ..., N-1. Plán obsahuje zoznam diaľničných úsekov (dvojíc miest) v poradí v akom budú dostavané. Predpokladáme, že na začiatku nie sú diaľnicou spojené žiadne dve mestá. Zaujímá nás po dostavaní koľkých úsekov bude diaľničná sieť spájať každé mesto s každým?



Úloha: Na vstupe (napr. v textovom súbore) máme počet miest a zoznam dvojíc popisujúci postupné dokončenie diaľničných úsekov. Každá dvojica zodpovedá jednému diaľničnému úseku spájajúcemu dvojicu miest. Vytvorte program, ktorý vypočíta, po dostavaní koľkých úsekov budú všetky mestá navzájom spojené diaľničnou sieťou.

Príklad pre N=5:

- 1 3 - diaľnica prepája mestá 1 a 3
- 0 3 - diaľnica prepája mestá 0 a 3, z mesta 0 sa vieme dostať aj do mesta 1 (cez mesto 3)
- 0 1 - diaľnica prepája mestá 0 a 1, táto diaľnica spája už spojené mestá
- 4 2 - diaľnica prepája mestá 4 a 2, teraz máme dva diaľničné komponenty 0, 1, 3 a 2, 4
- 3 2 - diaľnica prepája mestá 3 a 2, po dostavaní tohto úseku sú všetky mestá spojené

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

Návraty k midtermu (5 bodov)

Vieme, že platí toto tvrdenie: „Máme 2 usporiadané (neklesajúce) postupnosti (polia) A veľkosti n a B veľkosti m . Potom v čase $O(n+m)$ možno zistiť počet rôznych hodnôt s tou vlastnosťou, že sa nachádzajú v oboch postupnostiach, t.j. aj v A aj v B.“

Naprogramujte metódu, ktorá v uvedenom čase zistí počet rôznych hodnôt, ktoré sa nachádzajú v oboch poliach. Metóda nesmie modifikovať referencované polia. Akceptovaná pamäťová zložitosť je $O(1)$. Nie je dovolené používať triedy z JCF (napr. implementácie rozhrania Set).

```
public int velkostPrienu(int[] a, int[] b)
```

Príklad: $a = [1, 1, 2, 5, 5, 5, 9, 17, 20]$, $b = [1, 4, 5, 5, 8, 9, 9, 21, 21, 21]$

Metóda vráti hodnotu 3 lebo čísla 1, 5 a 9 sa nachádzajú v oboch poliach.