



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.

Ďaleký východ

Tohtoročné zadania z PAZ1b nás zatiaľ zaviedli na divoký západ, studený sever, suchý juh. Dnes sa pozrieme na ďaleký východ. Pomocou algoritmov budeme pomáhať turistom aj miestnym a riešiť každodenné aj exotické problémy.

Priveľká objednávka (18 bodov, backtracking)

Fabrika na výrobu tričiek v Bangladéši dostala objednávku na príliš veľa tričiek rôznych druhov. Aby nemuseli odmietnuť objednávku a stihli tričká vyrobiť včas, rozhodli sa, že objednávku rozdelia na menšie podľa druhov tričiek a každú malú objednávku presunú do inej fabriky.

Vieme, za koľko dní majú odovzdať všetky objednávky. O každej objednávke vieme počet tričiek. Jedna objednávka musí byť spracovaná v jednej fabrike. Ďalej o každej fabrike vieme koľko tričiek dokáže vyrobiť za deň, koľko stojí výroba jedného trička, koľko dní trvá príprava výroby a aká je cena nastavenia výroby na nový produkt.

Vzorový vstup (objednávky sú uvedené fialovou a ceny červenou):

8 (počet dní na odovzdanie objednávok)

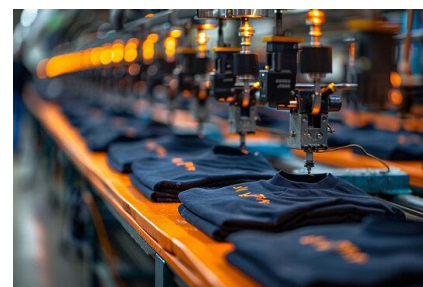
5 5 1 1 1 (počty tričiek v objednávkach)

1 1 7 3 (informácie o prvej fabrike *)

1 4 1 3 (informácie o druhej fabrike)

6 2 5 1 (informácie o tretej fabrike)

* = počet tričiek vyrobených za deň, cena za tričko, cena prípravy výrobnej linky, počet dní na začatie výroby



stock.adobe.com

Pozn.: Ak máme objednávku na 1100 tričiek a fabrika ich za deň vyrobí 1000, tak všetky tričká vyrobí za 2 celé dni. Teda pre jednoduchosť počítame v celých číslach.

Pozn.: Ako vypočítať aby $1100/1000$ bolo 2? Použite na to $(\text{int}) \text{Math.ceil}(1100.0 / 1000.0)$.

Riešenie: Prvá fabrika zoberie objednávku na 5, druhá fabrika zoberie objednávku na 1 a 1, tretia fabrika zoberie objednávku na 5 a 1. Prvá fabrika zvládne objednávku za $5/1+3=8$ dní za cenu $5*1+7=12$. Druhá fabrika zvládne objednávky za $1/1+3+1/1+3=8$ dní za cenu $1*4+1+1*4+1=10$. Tretia fabrika zvládne objednávky za $5/6+1+1/6+1=4$ dni, za cenu $5*2+5+1*2+5=22$.

Úlohy: (9b) Implementujte program, ktorý načíta za koľko dní majú byť všetky objednávky vyrobené. Načíta počet tričiek v každej objednávke. Ďalej načíta o každej fabrike koľko tričiek vyrobí za deň, aká je cena jedného trička, koľko dní trvá nastavenie výroby v danej fabrike a koľko stojí nastavenie výroby. Metóda vráti za najmenej koľko peňazí je možné spracovať všetky objednávky alebo -1 ak nie je možné ich za požadovaný počet dní spracovať. Formát vstupu si zvolte podľa svojich preferencií.

(+4b) Program pre najlacnejšie riešenie vypíše pre každú fabriku ktoré objednávky spracuje, koľko dní bude trvať ich spracovanie a aká bude cena ich spracovania.

(+3 až +5b) Podľa efektívnosti. Algoritmus je tým efektívnejší, čím viac eliminuje výpočty, ktoré nevedú k prípustnému riešeniu.

Choo choo (15 bodov, grafové algoritmy)

Cestovatelia v Japonsku častokrát využívajú cestovné pass-y na neobmedzené cestovanie v rámci časti železničnej siete. Táto sieť ponúka možnosť ísť pomalými lokálnymi vlakmi, diaľkovými vlakmi, až po Shinkanseny (rýchlovlaky). Lukáš, Peťo a Zoli sa rozhodli, že v jeden deň pobytu pôjdu vlakom tak ďaleko ako sa len v daný deň dá.



voyeglobal.com

Zobrali si dvojicu máp, každá mapa má tie isté mestá (vrcholy) a hrany (spojenia medzi nimi). Mapy sa líšia v tom, že jedna má na hranách vzdialenosti medzi mestami v km a druhá čas cesty medzi nimi v minútach.



jrailpass.com

Úlohy: (7b) Implementujte program, ktorý načíta mapy, čas aký majú na cestu a štartovacie mesto (vrchol). Tento program vráti do akého najvzdialenejšieho mesta (vrcholu) sa vedia zo štartu dopraviť do daného času.

(+4b) Za výpis cesty.

(+4b) Upravte svoj program tak, aby vypísal také najvzdialenejšie mesto, že sa za daný čas dostali do mesta a aj sa z neho stihli vrátiť.

Pozn.: Cesta hore kopcom a dole kopcom je rovnako dlhá ale trvá rôzne dlho. Napr. Ak ideme z A do B 1 minútu a z B do A 3 minúty, tak za 2 minúty nevieme ísť z A do B a naspäť z B do A.

SUMO! (13 bodov, dynamické programovanie)

Ak prídete do Japonska v správnom období, budete si môcť kúpiť lístky na turnaj v sumo. Lístky sa predávajú v rôznych mestách a sú na rôzny počet zápasov. O každom lístku viete deň prvého zápasu, deň posledného zápasu a počet zápasov ktoré uvidíte.

Vzorový vstup pre jeden lístok:

2 4 10 (začiatok 2. deň, koniec 4. deň, počet zápasov v lístku 10)

Každý lístok využijeme celý, teda pre vzorový lístok sme na turnaji v dňoch 2, 3 a 4, pričom uvidíme 10 zápasov. V jeden deň nemôžeme byť na turnajoch v dvoch rôznych mestách.



japanistry.com

Úlohy: (9b) Implementujte program, ktorý načíta zoznam lístkov ktoré sa predávajú a zistí najviac koľko zápasov v sumo môžeme vidieť.

(+4b) Vypíšte ktoré lístky je potrebné si zakúpiť aby sme videli čo najviac zápasov.

Ako nerobiť (12 bodov, greedy algoritmy)

Psychológovia prišli na to, že ľudia ktorí majú naplnené základné potreby ako jedlo, teplo a bezpečie, tak ku svojmu vyššiemu šťastiu potrebujú prácu ktorá ich naplňa, alebo robia niečo pre iných. Žiaľ v Severnej Kórei nemajú ľudia naplnené základné potreby, v tomto prípade sa ľudia nevenujú vyšším potrebám ako práca naplno a snažia sa robiť čo najmenej.



dailynk.com

Byun Kwon Park pracuje pri mixéri na mazadlo. Aby šetril svoju energiu a pracoval čo najmenej, chce pracovať iba vtedy keď naňho pozerá vedúci. Keďže v tejto krajine je všetko centrálné plánované, aj jeho vedúci má presné časy kedy ho príde skontrolovať a sú rovnaké každý deň.

Byun Kwon má zapísané časy kedy ho príde vedúci skontrolovať ako čas od začiatku pracovnej zmeny v minútach. Napr. 4, 6, 17, 31, 35, ... (môžete uvažovať utriedený vstup). Prišiel na to, že spracovanie dávky v mixéri mu prvýkrát trvá m minút, druhýkrát $m+p$ minút, tretí $m+2p$, štvrtý $m+3p$, ... lebo je čím ďalej tým viac unavený.

Pre časy kontroly 4, 6, 17, 31, 35 a $m = 3$, $p = 2$ prišiel na to, že stačí aby nahadzoval mixér trikrát, aby vždy počas kontroly pracoval.

Úlohy: (8b) Implementujte program, ktorý načíta časy kontrol (utriedený vstup), časy potrebné na spracovanie dávky v mixéri m a p . Vypíše minimálne koľko dávok v mixéri je potrebné spracovať aby vždy počas kontroly pracoval.

(+4b) Vypíšte časy kedy je potrebné začať pracovať v predchádzajúcej úlohe.

Čínska hierarchia (6 bodov, stromy)

(6b) Do triedy osoba z prednášky (nájdete v PDFku z prednášky alebo na Gitlabe), pridajte metódu `naRovnakejUrovni`, ktorá má jeden parameter `podlaKoho` a vráti list osôb. Tento list osôb obsahuje iba tie osoby, ktoré sú v podstrome osoby nad ktorou sa volala metóda a zároveň sú na rovnakej úrovni ako osoba zadaná parametrom `podlaKoho`. Napr. Ak je `podlaKoho` vnuk, tak metóda vráti všetkých vnukov v podstrome.

Pozn.: Ak potrebujete, rozdeľte si úlohu na viacero metód.

```
public List<Osoba> naRovnakejUrovni (Osoba podlaKoho)
```

China's leadership

