

Korešpondenčný seminár z programovania
XXVIII. ročník, 2010/11
Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

KSP finančne podporujú: MICROSTEP-MIS spol. s r.o.

Agentúra na podporu výskumu a vývoja

Príklady 2. kola zimnej časti

Milé naše riešiteľky, milí naši riešitelia,

prajeme Vám veľa zábavy pri riešení 2. série nášho korešpondenčného seminára. Dúfame, že sa Vám úlohy budú páčiť aspoň tak ako tie z 1. série. Dovoľme si ešte pripomenúť, že pre tých, ktorým bežné úlohy nestačia je tu kategória T, ktorej víťaz získa knihu podľa vlastného výberu (a kopolu hodnotných skúseností navyše).

Termín odoslania riešení tejto série je pondelok **10. januára 2011.**

Opakovanie je matkou múdrosti a otcom sériovej výroby.

KSPáci

1. Zmiešaná postupnosť

kat. Z; 10 bodov

Tánička veľmi rada triedi čísla. Lenže keď nemá čo triediť, tak sa nudí, a keď sa nudí, vyžiera ostatným KSPákom desiatu. Keďže KSPáci sú veľmi neradi hladní, tak sa rozhodli, že ju musia nejako zabaviť. Povedali si, že jej budú podhadzovať postupnosti. Aby to malo efekt, potrebujú, aby dané postupnosti **neboli** utriedené. Bohužiaľ, nemajú čas to kontrolovať (musia si strážiť desiatu), a preto vás žiadajú o pomoc.

Úloha: Na prvom riadku vstupu je dané číslo N (dĺžka postupnosti, $N \geq 3$) a na druhom nasleduje postupnosť N navzájom rôznych čísel oddelených medzerou. Napíšte program, ktorý túto postupnosť preusporiada a vypíše na výstup tak, aby nebola rastúca ani klesajúca.

Príklad:

Vstup:

5
1 2 3 4 7

Výstup:

4 7 2 3 1
Vyhovuje ľubovoľné poradie okrem 1 2 3 4 7
a 7 4 3 2 1.

2. Zase tie závislosti

kat. Z; 10 bodov

Tento príklad bude o Linuxe, balíčkoch a inštalovaní. Ak chceme v Linuxe nainštalovať nejaký program, musíme nainštalovať balíček, ktorý ho obsahuje. Avšak program potrebuje pre svoj beh napríklad knižnicu. Každá knižnica je v nejakom balíčku a preto potrebujeme mať nainštalované všetky balíčky s knižnicami, ktoré potrebuje náš program. Tomuto hovoríme závislosť.

Ku každému balíčku v systéme poznáme zoznam balíčkov, na ktorých závisí. Tie však tiež potrebujú mať nainštalované nejaké ďalšie balíčky a tie ešte ďalšie balíčky (a takto by sme mohli pokračovať niekoľko strán).

Aby sme to nemali také ťažké, existujú rôzne pomocné programy. Povieme im zoznam balíčkov, ktoré chceme nainštalovať, a oni okrem nich nainštalujú aj všetky ostatné potrebné balíčky.

Najlepšie to demonštrujeme príkladom. Nech balíček A závisí na balíčku B , B na C a E , C na D , E na F a G . Ak necháme nainštalovať balíček A , tak sa nainštalujú aj balíčky B, C, D, E, F a G , lebo na nich priamo alebo nepriamo závisí program A .

Niekedy potrebujeme „naklonovať“ nainštalované balíčky na druhý počítač. Môžeme si spraviť zoznam všetkých balíčkov, ktoré sú nainštalované na prvom počítači, a presne tie nainštalovať na druhý počítač. Avšak ako vidno z príkladu vyššie, stačí nainštalovať iba niektoré balíčky a zvyšné sa doplnia automaticky.

Úloha: Prvý riadok vstupu obsahuje číslo N – počet balíčkov, ktoré chceme nainštalovať. Nasleduje N riadkov, i -ty z nich popisuje i -ty balíček. Každý taký riadok začína počtom balíčkov, na ktorých závisí balíček i , a pokračuje ich zoznamom. Všetky balíčky sú reprezentované celými číslami od 1 po N .

Vašou úlohou je napísať program, ktorý nájde a vypíše najmenšiu sadu balíčkov, ktorej nainštalovanie spôsobí nainštalovanie všetkých balíčkov.

Môžete predpokladať, že v závislostiach medzi balíčkami nie je cyklus a že balíčkovací systém je korektný (každý balíček závisí len na existujúcich balíčkoch). Cyklus v závislostiach je taká postupnosť balíčkov b_1, b_2, \dots, b_k , že balíček b_1 závisí na b_2 , b_2 na b_3 , \dots , b_{k-1} na b_k a b_k závisí na b_1 .

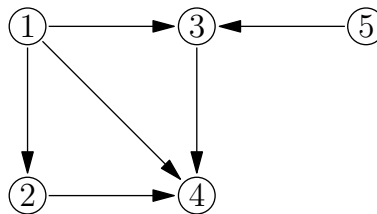
Príklad:

Vstup:

```
5
3 2 3 4
1 4
1 4
0
1 3
```

Výstup:

```
1 5
Takto vyzerajú závislosti balíčkov zo vstupu:
```



3. Zábudlivý dôchodca

kat. Z; 10 bodov

V dobe krutej finančnej krízy sa snaží každá továreň minimalizovať svoje náklady na prevádzku, preto sa prijímajú aj lacnejšie pracovné sily z radov dôchodcov, študentov a nelegálnych prisťahovalcov. Inak tomu samozrejme nie je ani v továrni na triedenie čísiel. Preto keď prišla ponuka pána Michala, že bude otročiť za jedlo, šaty a vzduch, bolo vedenie nadšené.

Rýchlo ale zistili, že to má drobný háčik. Pán Michal škúli, takže nikdy nevidí dve susedné čísla. To by ešte nebolo také hrozné, kebyže neprišiel pri absurdnej nehode s kapustou a zbijačkou o krátkodobú pamäť. Takže príslovie *zíde z očí, zíde z mysle* preňho bohužiaľ platí doslovne.

Majiteľ však zistil, že keď nechá pána Michala triediť niektoré zákazky, napriek svojmu handicapu ich utriediť dokáže. Chcel by mať program, ktorý by mu pri zadaní zákazky povedal, či ju môže dať na starosť pánovi Michalovi. No a keď už je tá kríza, a vy ste študenti. . . môžete hádať.

Úloha: Na vstupe máte dané číslo N , v ďalšom riadku nasleduje N celých čísiel. Vašou úlohou je rozhodnúť, či sme schopní získať z týchto čísiel neklesajúcu postupnosť, ak v jednom kroku môžeme vymeniť iba dvojicu čísel na pozíciách k a $k + 2$ ($1 \leq k \leq N - 2$).

Príklad:

Vstup:

```
5
10 45 52 14 37
```

Výstup:

```
Ano.
Vymeníme najprv čísla na treťom a piatom
mieste, potom tie na druhom a štvrtom mieste.
```

Príklad:**Vstup:**

4
1 3 2 4

Výstup:

Nie.
Výmenou prvého a tretieho, ani výmenou druhého a štvrtého čísla si veľmi nepomôžeme.

..... Tu sa začínajú zadania KSP-O

4. Zákerné kopce

kat. Z a O; 15 bodov

Rytier Artur sa zúfalo pozerá pred seba. Ten svah, čo má pred nosom, sa mu vôbec, ale vôbec nepáči. Je strašne strmý – keď sa na neho pokúša vyškríabať, vždy sa zošmykne späť dole. On sa ale naozaj potrebuje dostať na ten kopec. Hmm, že by to šlo nejak okolo?

Byť rytierom nie je žiadna slasť, také ťažké brnenie vám umožňuje vyškrabať sa len na svah, ktorý na jednom yarde nestúpne o viac ako K stôp. Našťastie je Artur statný a vysoký a vie nejaký ten jarok aj preskočiť, pokiaľ nemá viac ako yard.

Terén si môžeme zjednodušené predstaviť ako štvorcovú sieť s R riadkami a C stĺpcami, kde poznáme výšku každého políčka. Artur sa vie hýbať na jeden krok/skok len v štvorci 5×5 okolo seba (čiže jedným skokom vie každú svoju súradnicu zmeniť maximálne o 2). Súčasne pri žiadnom kroku/skoku nemôže prekonať výškový rozdiel viac ako K . Podarí sa mu najšľ cestu?

Úloha: Prvý riadok vstupu obsahuje celé čísla R, C – výšku a šírku mapy – a celé číslo K , ktoré určuje najväčší rozdiel (či už stúpanie alebo klesanie), cez ktorý vie Artur v jednom kroku/skoku prejsť. Druhý riadok obsahuje celé čísla r_0, c_0, r_F, c_F – súradnice začiatočného políčka (r_0, c_0) a koncového políčka (r_F, c_F) . Ďalej nasleduje R riadkov, v každom z nich sa nachádza C celých čísel – výška daného políčka mapy.

Pre rozmery mapy bude platiť $1 \leq R, C \leq 1000$. Ľavý horný roh mapy má súradnice $(1, 1)$ a pravý dolný roh zasa (R, C) . Súradnice začiatočného a koncového políčka budú vnútri mapy. Výšky políčok a rovnako aj hodnota K sú z rozsahu $[0, 10^9]$ vrátane.

Vašou úlohou je vypísať dĺžku najkratšej cesty z políčka (r_0, c_0) na (r_F, c_F) , alebo reťazec „Artur, mas smolu.“ (bez úvodzoviek), ak cesta neexistuje.

Odovzdávanie: Tento príklad je praktický, čo znamená, že sa dá odovzdať jedine elektronicky, pričom stačí odovzdať iba zdrojový kód. Pri odovzdaní sa vám program otestuje na niekoľkých vstupoch a dostane body podľa toho, na koľkých vstupoch dá správny výsledok v časovom limite.

Príklad:**Vstup:**

10 10 3
1 1 10 10
2 0 0 0 0 0 0 0 0 0
0 0 0 8 0 0 0 0 0 0
0 5 0 0 0 0 0 0 0 0
0 0 0 0 0 6 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 4 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 7 0
0 0 0 0 0 0 0 0 500 2000
0 0 0 0 0 0 0 0 100 9

Výstup:

6
Trasa vedie po políčkach s výškami 2, 5, 8, 6, 4, 7, 9. Všimnite si, že Artur nedoskočí z políčka s výškou 5 na políčko s výškou 6, lebo je ďaleko. Zároveň si všimnite, že Artura pri skoku zaujíma iba štartovacia a koncová výška, nie výška políčok medzitým.

5. Ošemetná situácia

kat. Z a O; 15 bodov

Dominika si pri každom nákupe v blízkom hypermarkete kúpi pár škatúľ mlieka. Potom si ich pekne uloží do poličky. Naľavo tie, ktoré treba urýchlene spotrebovať, a napravo tie, ktoré ešte nejaký čas počkajú.

Žiaľ, po poslednej oslave nič neostalo na svojom mieste a mlieka sa záhadne pomiešali. Rada by ich dostala do pôvodného stavu, no nevie ako na to, pretože na intráku je málo miesta a nemôže ich všetky naraz vybrať von.

Úloha: Prvý riadok vstupu obsahuje jediné celé číslo N , ktoré označuje počet škatúľ mlieka v poličke. Na druhom riadku je N čísiel (každé z nich je buď 1, 2 alebo 3) označujúcich trvanlivosť danej škatule.

Dominika môže naraz vymeniť len dve mlieka. Vypíšte takú postupnosť výmen, že po jej vykonaní budú škatule vo vzostupnom poradí podľa trvanlivosti.

Kvalita vášho algoritmu bude hodnotená podľa nasledovných kritérií (zoradené podľa dôležitosti):

1. počet výmen (na toto sa pozeráme z pohľadu O-notácie, takže dva algoritmy, čo spravia napríklad $5N^2$ a $3N^2 + 47N$ výmen, sú rovnako dobré),
2. časová zložitosť,
3. pamäťová zložitosť.

Príklad:

Vstup:

3
3 3 2

Vstup:

4
2 3 2 1

Výstup:

1 3

Výstup:

1 4
2 4

..... Tu sa končia zadania KSP-Z

6. O švédskych stoloch

kat. O; 20 bodov

Kde bolo, tam bolo, v škole v prírode bolo raz jedno sústredenie. Toto sústredenie však bolo špeciálne – nachádzali sa na ňom hory jedla umiestnené na švédskych stoloch.

Švédske stoly fungujú tak, že keď k stolu príde účastník sám, naloží si množstvo jedla ekvivalentné jeho hladu. Keď však príde k stolu ďalší účastník, tak rovnako ako aj pôvodný, rozhodne nechce zaostať a priloží si navyše. Keď príde viacero účastníkov, tiež nechce nikto hladovať, každý chce ostatných tromfnúť a tak si za každého priloží.

Hora jedla je však predsa len obmedzená, z čoho vyplýva, že vedúci musia riešiť závažný problém: kto dostane jedlo?

Úloha: Na vstupe je v prvom riadku počet účastníkov N a množstvo dostupného jedla M . Nasleduje N riadkov v tvare $x_i y_i$, kde x_i určuje množstvo jedla, ktoré skonzumuje i -ty účastník a y_i určuje množstvo jedla, ktoré i -ty účastník zje navyše za každého ďalšieho účastníka pri stole. Ak teda vedúci k stolu pustia k účastníkov, i -ty účastník zje $x_i + (k-1) \cdot y_i$ jedla. Napíšte program, ktorý zistí, koľko najviac ľudí sa môže najesť.

Príklad:

Vstup:

5 47
7 5
10 9
3 17
15 3
0 42

Výstup:

2
Vedúci môžu pustiť k stolu napríklad prvého a tretieho účastníka.

7. Oktávy

kat. O; 20 bodov

Halucinka robí rada na sústredkách budičky gitarou. Otvorí si svoj spevník a začne spievať nejakú melódiu. Tým zobudí Hermiho, ktorý sa chce hneď pridať. Avšak polorozospatý Hermi nezačne spievať presne to, čo Halucinka, ale svoju melódiu. Halucinka rozmýšľa, či Hermi vlastne spieva tú istú pieseň. Pomôžte Halucinke zistiť, či Hermi spieva v inej tónine, alebo je natoľko rozospatý, že spieva inú pieseň.

Úloha: Na vstupe dostanete popis dvoch melódií – Halucinkinej a Hermiho. Každý popis začína číslom vyjadrujúcim počet tónov danej melódie, nasledujú jednotlivé tóny. Hermiho melódia je nanajvýš taká dlhá ako Halucinkina.

Halucinka aj Hermi majú rozsah 3 oktávy, dokážu preto zaspievať tieto tóny (v poradí podľa výšky): C0, Cis0, D0, Dis0, E0, F0, Fis0, G0, Gis0, A0, Ais0, H0, C1, Cis1, D1, ..., Ais2, H2, C3.

Napíšte program, ktorý zistí, či si Hermi spieva tú istú pieseň ako Halucinka, teda či sa dá jeho melódia transponovať tak, aby vytvorila motív (čítaj podreťazec) Halucinkinej. Transponovať znamená každý tón Hermiho melódie posunúť o rovnaký počet poltónov (rovnakým smerom).

Riešenia, ktoré sa spoliehajú na generátor náhodných čísel a bez neho by dosahovali horšiu časovú zložitosť, nemajú šancu získať plný počet bodov. Pri výpočte časovej a pamäťovej zložitosti vášho algoritmu nezanedbávajte počet tónov T v Halucinkinom a Hermiho rozsahu ako konštantu (aj keď vieme, že $T = 37$).

Príklad:

Vstup:

24

C1 D1 E1 F1 F1 F1 F1 E1 D1 E1 E1 E1 E1 D1 C1 D1 D1 D1 D1 E1 D1 C1 C1 C1

12

H0 Ais0 Gis0 Ais0 Ais0 Ais0 Ais0 Gis0 Fis0 Gis0 Gis0 Gis0

Výstup:

Ano

Halucinka si spieva Kohútika jarabého a Hermi tiež, na vstupe je zachytený jeho prostriedok. Celý Hermiho Kohútik jarabí (posunutý o 6 poltónov nižšie) znie:

*Fis0 Gis0 Ais0 H0 H0 H0 H0 Ais0 Gis0 Ais0 Ais0 Ais0 Ais0 Gis0 Fis0 Gis0 Gis0 Gis0 Gis0
Ais0 Gis0 Fis0 Fis0 Fis0.*

Príklad:

Vstup:

20

C1 D1 E1 F1 G1 F1 F1 F1 D1 C1 C1 D1 E1 F1 G1 F1 F1 F1 D1 C1

12

A1 H1 Cis2 Cis2 H1 A1 E2 E2 Fis2 E2 H1 H1

Výstup:

Nie

Halucinka si spieva Na zelenej lúke sedí zajac a Hermi si spieva Pod horou, pod horou.

8. Obedy sa vydávajú do siedmej

kat. O; 25 bodov

Na intrákoch prednávkou otvorili novú Novú jedáleň. Aby prežila v tvrdej konkurencii, pripravujú v nej pomocou moderných technológií jedlo na mieru. Najprv každý študent podstúpil síce nepríjemnú, ale zato povinnú analýzu svojich chuťových buniek; na základe jej výsledkov mu potom priradili jedinečný pokrm. Po prekonaní prvotnej nedôvery ale študenti zistili, že obedy na mieru sú naozaj vynikajúce a čakať v rade na ne sa oplatí.

Všetko jedlo pripravuje automatizovaná výrobná linka, na ktorej konci stojí zamestnan-kyňa jedálne (študentmi nesprávne označovaná ako *teta kuchárka*) a vydáva obedy. Výrobná

linka je však automatizovaná až príliš. Každé ráno si sama vymyslí a vytlačí zoznam N jedál, ktoré v ten deň pripraví presne v poradí, v akom sa nachádzajú na zozname (jedlá na ňom sa môžu aj opakovať).

Teta kuchárka sa smutne pozerá na dnešný rad M študentov, ktorí čakajú na obed. Ich poradie sa totiž podstatne líši od zoznamu jedál, ktorý na dnes vytlačila výrobná linka. No a pretože študenti odmietajú jesť obed pripravený pre niekoho iného, často stojí teta kuchárka pred dilemou: vyhodíť porciu alebo študenta?

Pomôžte tete kuchárke a nájdite postup, ktorý zaručí, že sa naobedeje čo najviac študentov.

Úloha: V prvom riadku vstupu sú zadané čísla M a N ($1 \leq M, N \leq 100\,000$). Druhý riadok reprezentuje rad študentov: čísla a_1, a_2, \dots, a_M . Tretí riadok zase obsahuje zoznam jedál, ktoré dnes pripraví výrobná linka: čísla b_1, b_2, \dots, b_N . Platí $1 \leq a_i, b_i \leq 10^9$. V rade študentov sa žiadne číslo neopakuje. Všetky čísla na riadku sú oddelené jednou medzerou.

Nájdite a na jediný riadok výstupu vypíšte najväčšie také číslo K , že sa zo zoznamu jedál aj z radu študentov dá vyškrtáť niekoľko vhodných čísel tak, aby nám zostali dve rovnaké postupnosti dĺžky K (tí študenti, ktorí dnes dostanú svoju porciu).

Odovzdávanie: Tento príklad je praktický, čo znamená, že sa dá odovzdať jedine elektronicky, pričom stačí odovzdať iba zdrojový kód. Pri odovzdaní sa vám program otestuje na niekoľkých vstupoch a dostane body podľa toho, na koľkých vstupoch dá správny výsledok v časovom limite.

Príklad:

Vstup:

5 8
7 3 4 8 1
2 2 3 9 1 8 4 1

Výstup:

3
Naobedujú sa študenti 3, 4, 1 alebo 3, 8, 1.

..... Tu sa končia zadania KSP-O a začínajú zadania KSP-T

9. Tesnotka pred kontrolou

kat. T; 25 bodov

Po neúspechu s predajom slúchadiel sa Bob rozhodol, že si otvorí kníhkupectvo. Doba je ale ťažká a tak sa rozhodol, že popri predaji legálnej literatúry si bude privyrábať aj predajom tej ilegálnej, zaoberajúcej sa programovaním.

Čo čert nechcel, kontrola z EÚ sa rozhodla navštíviť Bobovo kníhkupectvo. Keď to zistil, mal práve v obchode párne počet ilegálnych kníh a dvoch zákazníkov, ktorí boli zaujatí týmto tovarom. Rozhodol sa, že využije túto príležitosť, zbaví sa kníh a rozdelí im ich rovným dielom. Každý zákazník si po kúpe spočíta, koľko v priemere zaplatil za jednu samohlásku¹. Bob chce kvôli budovaniu dobrého mena firmy dosiahnuť, aby bol súčet týchto dvoch hodnôt čo najmenší.

Úloha: Prvý riadok vstupu obsahuje párne číslo N ($2 \leq N \leq 50$). Druhý riadok obsahuje N medzerou oddelených čísel, pričom k -te číslo vyjadruje počet samohlások v k -tej knihe. Tretí riadok obsahuje N medzerou oddelených čísel, pričom k -te z nich zasa vyjadruje cenu k -tej knihy. Počty samohlások aj ceny kníh sú celé čísla z rozsahu $[1, 500]$ (vrátane).

Vašou úlohou je rozdeliť tieto knihy na dve kopy tak, aby v každej kope bolo $N/2$ kníh a keď spočítame priemernú cenu za samohlásku v každej kope, tak súčet týchto dvoch hodnôt bude čo najmenší.

Na výstup vypíšte jedno číslo: najmenší možný dosiahnuteľný súčet priemerných cien. Vypíšte aspoň 10 desatinných miest. Vaša odpoveď by mala mať absolútnu alebo relatívnu chybu najviac 10^{-9} .

¹to je predsa to, čo nás v tejto literatúre zaujíma

Odvzdávanie: Tento príklad je praktický, čo znamená, že sa dá odovzdať jedine elektronicky, pričom stačí odovzdať iba zdrojový kód. Pri odovzdaní sa vám program otestuje na niekoľkých vstupoch a dostane body podľa toho, na koľkých vstupoch dá správny výsledok v časovom limite.

Príklad:

Vstup:

2
2 4
3 4

Vstup:

4
1 1 2 2
1 1 1 1

Výstup:

2.5

Výstup:

1.3333333333
Každému zákazníkovi Bob predá knihu s jednou a s dvoma samohláskami.

10. Turbulencia a krevety

kat. T; 25 bodov

Tomi a Jano sedia v lietadle. Let je dlhý a preto je jeho súčasťou aj obed. Letecká spoločnosť pridala k obedu špeciálny bonus – originál krevety v plastovom tégliku. Väčšina cestujúcich si ich len nedôverčivo obzrela, ale niektorí odvážlivci sa rozhodli vyskúšať túto novotu. Lenže chvíľu nato prišla silná turbulencia, viaceré žalúdky spravili piruetu a patriční odvážlivci okrevetili celé svoje široké okolie.

Právnici letovej spoločnosti zistili, že nie sú viazaní preplácať škodu všetkým – stačí, keď ako symbolické gesto vynahradia škodu tomu, kto je na tom najhoršie, aby nevyzerali zle v novinách.

Tzv. *najväčší chudák* je človek, ktorého zasiahlo svojimi krevetami najviac ľudí, ale pritom si sám krevety nedal a teda za nič nemôže. Vašou úlohou je zistiť, kto to je a koľko cestujúcich ho zasiahlo (lebo od toho potom záleží, koľko mu treba zaplatiť).

Úloha: Lietadlo je dlhé R radov a každý rad je široký S sedadiel. Na každom sedadle sedí jeden cestujúci. N cestujúcich si dalo krevety a potom okrevetili svojich okolosediacich – konkrétne štvorec so stranou D , v strede ktorého sú. (D je nepárne číslo.)

Na vstupe sú prirodzené čísla R , S , N a D ; ďalej N riadkov so súradnicami jednotlivých odvážlivcov (súradnice sú v poradí rad, sedadlo; čísloje sa od 1). Aspoň jeden cestujúci si krevety nedal. Váš program má vypísať jedno číslo – koľko ľudí zasiahlo najväčšieho chudáka.

Lietadlo je interkontinentálne a teda strašne veľké. Zložitosť vášho programu nesmie závisieť od R a S , lebo to bude trvať večnosť a leteckú spoločnosť medzitým všetci novinári roznesú v zuboch.

Príklad:

Vstup:

6 5 3 5
2 1
4 3
6 5

Výstup:

2
Cestujúceho na sedadle (4, 3) síce okrevetili traja, ale keďže si sám dal krevety, nemôže to byť najväčší chudák.