



Korešpondenčný seminár z programovania XXIII. ročník, 2005/06

Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

*KSP finančne podporujú: aSc – Applied Software Consultants spol. s r.o.
MICROSTEP-MIS spol. s r.o.*

Príklady 2. kola letnej časti

Milé naše riešiteľky, milí naši riešitelia,
leto sa nám už blíži, pomaly nás dobieha, tak na skrátenie dlhých chvíľ sme pre Vás pripravili úplne novú sériu vášho najobľúbenejšieho korešpondenčného seminára. Takže sa pustite do riešenia týchto pekných príkladov. A nahovorte aj kamarátov. Jesenné sústredko za tú trochu námahy určite stojí :-).

Svoje riešenia nám posielajte najneskôr do **15. mája 2006**. A samozrejme, ak chcete dostať nejakú poštu, tak nám pošlite aj známky v hodnote rádovo 15 Sk.

Komu niet rady, tomu niet postupnosti...

KSPáci

1. O guľičkách a tuneloch

15 bodov

Raz si Marek doma upratoval a našiel guľičky. A nie také obyčajné, ale dokonca aj farebné. A potom našiel rúrky. Také, do ktorých sa zmestila každá guľička. Tak zobral lepidlo a začal stavať. Postavil jedno ohromné dielo. Samé prepojené rúrky. Navrchu bol lievnik, kde sa hádzali dnu, na spodku miska kam vypadávali. A navyše to postavil tak, že sa dalo prejsť každou rúrkou. A začal zhora hádzať guľičky.

V diele bolo veľa rozdvojek, roztrojiek a mnoho iných križovatiek, guľičky chodili po rôznych cestách... až Mareka začalo trápiť svedomie, že si tu hádže guľkami namiesto robenia úlohy z Úvodu do kombinatoriky a teórie grafov. Aby svedomie odbil, rozhodol sa, že zistí, koľko existuje možných ciest z vrchu až úplne nadol. Ale už bol neskorý večer, a tak vás požiadal o pomoc.

Úloha: Na vstupe sú dve prirodzené čísla N a M . N označuje počet uzlov, M označuje počet rúrok. Potom nasleduje M popisov jednotlivých rúrok. Na každom riadku budú dve čísla a_i, b_i , pričom a_i určuje, z ktorého uzla vedie i -ta rúrka a b_i určuje, kam daná rúrka vedie. Guľička nevie ísť v opačnom smere¹ (čiže z b_i do a_i). Všeobecnejšie, môžete predpokladať, že ak sa dá cez rúrky dostať z uzla u do uzla v , tak sa nedá dostať z uzla v do uzla u . Vašou úlohou bude vyrátať, koľko existuje rôznych ciest z uzla 1 do uzla N .

Príklad:

Vstup:

$N = 7, M = 8$

1 2

1 6

2 4

2 5

6 3

4 7

5 7

3 7

Výstup:

3

(Cesty:

$1 \rightarrow 6 \rightarrow 3 \rightarrow 7$,

$1 \rightarrow 2 \rightarrow 4 \rightarrow 7$,

$1 \rightarrow 2 \rightarrow 5 \rightarrow 7$)

¹jedine, že by niekto gravitáciu obrátil...

2. Odíd' vírus

15 bodov

Kto už na tomto svete nepočul o počítačových vírusoch – kúskoch kódu, ktoré sa vedia pripojiť k spustiteľným súborom. Keď sa daný súbor spustí, spustia sa aj oni a vedia narobiť veľké škody. Pri podozrení, že sa v počítači nachádza vírus, je potrebné prehľadať všetky súbory, vírusy nájsť a odstrániť. Vašou úlohou bude naprogramovať taký veľmi jednoduchý antivírus.

Úloha: Na vstupe dostanete číslo N a N postupností núl a jednotiek v_1, \dots, v_N (popisy jednotlivých vírusov). Ďalej nasleduje S – ďalšia postupnosť núl a jednotiek (skúmaný súbor). Pre každý vírus v_i vypíšte všetky jeho výskyty v súbore S alebo správu, že daný vírus sa v súbore nenachádza. Môžete predpokladať, že kód žiadneho vírusu nie je obsiahnutý v kóde nejakého iného – t.j. žiadne v_i nie je podreťazcom (súvislou podpostupnosťou) nejakého v_j pre $j \neq i$.

Príklad:

Vstup:

$N = 3$

0011, 010, 10110

$S = 001101010011001$

Výstup:

0011 sa nachádza na pozíciách 1 a 9

010 sa nachádza na pozíciách 5 a 7

10110 sa v súbore nenachádza

3. O podenkách

15 bodov

Pri sťahovaní KSP sme objavili v zabudnutom rohu i33 akvárium, kde žili podenky. Podenky sú taký zvláštny druh vodného hmyzu – žijú iba pár hodín, počas ktorých nakladú vajíčka a neskôr zahynú. Z týchto vajíčok sa v ďalšej generácii vyľahnu podenky, ktoré žijú tiež len veľmi krátko a cyklus pokračuje...

Keďže v novej miestnosti je málo miesta, akvárium budeme musieť vyhodiť. Zároveň sme však mierumilovní a nechceme zabiť ani jednu podenku. Pomôžte nám zistiť pravdepodobnosť, že po M generáciách naše podenky vyhynú.

Úloha: Na vstupe sú čísla K a N . K je aktuálny počet podeniiek v akváriu. Ďalej nasleduje N hodnôt – pravdepodobnosti p_0, p_1, \dots, p_{N-1} . Pravdepodobnosť p_i určuje, s akou pravdepodobnosťou jedna podenka nakladie i vajíčok. Súčet všetkých p_i je samozrejme 1. Nakoniec je na vstupe číslo M . Vašou úlohou je zistiť pravdepodobnosť, že po M generáciách nebude žiť žiadna podenka.

Príklad:

Vstup:

$K = 1, N = 3$

0.33

0.34

0.33

$M = 2$

Výstup:

0.478

4. O duplikátoch a pánovi Dirichletovi...

15 bodov

„Viac vľavo, nie, to bolo príliš, trochu naspäť vpravo.“

„Je to už dobré?“

„No, ešte to posuň trochu hore... správne. Teraz je to dobre.“

„Super, daj ďalší plagát. Počkaj, ale to je IOI 2000, to som už niekde lepil...“

„Ľala, veď tam visí na stene...“

„Ale ako je to možné? Žeby sme mali viac rovnakých plagátov?“

„A koľko je vlastne všetkých?“

„ $N + 1$ “

„Tak veľa?“

Problém je totiž nasledovný. KSPáci sa presťahovali do novej miestnosti a začali si tam lepiť plagáty². Ale zistili, že nejaké sa opakujú. Ako tak začali nad tým maturovať, tak im napadla nasledovná úloha a uvedomili si, že by to bola zaujímavá úloha do KSP. Tak teda, tu ju máte...

Úloha: Máte číslo N a pole $A[0..N]$, v ktorom sú uložené čísla od 1 po N . Nieкто by povedal³, že keďže dĺžka poľa je $N + 1$ a čísel je len N , nachádza sa aspoň jedno číslo v poli (aspoň) dvakrát. Vašou úlohou je nájsť jedno takéto číslo. Ak je ich viac, stačí ľubovoľné z nich. Jediné obmedzenie, ktoré máte, je, že pole nesmiete prepisovať.

Riešenie je tým lepšie, čím je menší súčin jeho časovej a pamäťovej zložitosti. (Pod pamäťou sa myslia pomocné premenné okrem samotného poľa.) Úplne najlepšie je teda riešenie v lineárnom čase, ktoré si vystačí s konštantným počtom pomocných premenných.

Príklad:

Vstup:

$N = 5$

$A = (1, 3, 2, 4, 1, 2)$

Výstup:

1

5. O tabuľke profesora Kolmogorova

15 bodov

Na stole profesora Kolmogorova ležala tabuľka, husto popísaná číslami. Vyzerala asi takto:

x	f(x)	x	f(x)	x	f(x)	x	f(x)
1	47	13	68	125	90	47300	171957387
2	47	14	65	126	103	47400	5975472
3	47	15	66	970	2544	789234	11737933
4	47	16	79	1003	2734	789235	11739988
5	47	17	85	1005	2719	789236	11742111
6	47	18	91	7410	146055	789237	11744214
7	47	19	89	7411	146035	1234567	32762148
8	71	20	103	15233	617165	2345678	12322037
9	72	47	302	35233	95383914	34567890	27469977
10	77	74	701	35234	95389328		
11	78	123	90	36233	100878311		
12	67	124	90	36234	100883886		

„Andrej Nikolajevič, čože to máte za tabuľku?“ zaujímala sa jeho diplomantka Nataša.

„Ále, to nič nie je...“ zamrmľal roztržito profesor, no po chvíli dodal:

„Ako dlho by vám trvalo zapamätať si ju celú?“

Nataša chvíľku hľadela na tabuľku, a potom odpovedala: „Tolkoto.“

Profesor uznanlivo pokýval hlavou.

Úloha: Natašin „trik“ spočíval v tom, že pochopila, ako profesorova tabuľka vznikla. Skús to aj ty. Napiš **čo najkratší** program, ktorý bude počítat funkciu uvedenú v tabuľke. (Je úplne jedno, aký výsledok vráti pre iné hodnoty x ako tie uvedené v tabuľke.)

Vstupom pre tvoj program bude súbor obsahujúci presne 470 riadkov. V každom z nich je jedno z čísel, ktoré sú uvedené v tabuľke v stĺpci „x“. Výstup tvojho programu musí mať presne 470 riadkov, pričom ak označíme číslo v i -tom riadku vstupu x_i , tak v i -tom riadku výstupu musí byť číslo $f(x_i)$.

Programy píšete v Pasmale, C alebo C++ a odovzdávajte ich v našej Programátorskej liahni ako riešenie príkladu „Tabuľka profesora Kolmogorova“ (nájdete ju v piatej sade úloh, po prihlásení je URL https://liahen.ksp.sk/task.php?task_id=40). Pri hodnotení do

² „Na stromy aj na chaty, všade lepím plagáty!“

³ napríklad pán Dirichlet

KSP nezáleží na voľbe programovacieho jazyka, pokúsime sa, aby to isté riešenie napísané v Pascale aj v C dostalo do KSP rovnako bodov.

Neposielajte nám papierové riešenie tohto príkladu! Hodnotiť sa budú len programy, ktoré náš automatický testovač vyhodnotí ako naozaj správne.

Príklad:

Vstup:

15
3
123
...

Výstup:

66
47
90
...