Korešpondenčný seminár z programovania
XXII. ročník, 2004/05
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

Tu tú, di du du
Tu tú du du
Tu tú di du du di du du di du du dutú tu du tu
Mahna Mahna
M-ahna M-ahna ma **2. mája 2005** M-ahna mana mana ma M-ahna M-ahn...

– The question is, what is a Mahna Mahna? – The question is, who cares?

KSPáci

1. O aritmetickej postupnosti

15 bodov

„Aha, je tam: 2 6 10“ „Ale tá nemá správnu diferenciu.“ „Tak dobre, tak tam nie je.“ – Ozýva sa hádka Paľa a Moniky z prednáškovej miestnosti. Aspoň, že prednášajúci ešte neprišiel. Už by im dávno vynadal, nech si napíšu program, ktorý to zráta, aby sa nemuseli hádať.

Úloha:

Na vstupe sú čísla A a N. Nasleduje N čísel. Vašou úlohou je zistiť, či sa dá spomedzi nich vybrať niekoľko tak, aby (nie nutne v poradí, v akom sú na vstupe) tvorili aspoň trojčlennú aritmetickú postupnosť. Navyše musí platiť, že ak ste vybrali k čísel, tak diferencia výslednej postupnosti musí byť $\frac{A}{(k-1)}$. Teda ak si napr. myslíte, že je tam aritmetická postupnosť dĺžky 5, tak musí mať diferenciu $\frac{A}{4}$.

Ak to náhodou neviete, tak n-členná aritmetická postupnosť s diferenciou d je postupnosť $a_0, a_0 + d, a_0 + 2d, \ldots, a_0 + (n-1) \cdot d$.

Vstup:	**Výstup:**
$A = 9$, $N = 5$ 11 5 9 14 8	áno *($\{5, 8, 11, 14\}$ je 4-členná postupnosť s diferenciou $A/3 = 3$)*

2. O štatistikárení

15 bodov

Xanadu, Xanadu, (now we are here)

In Xaaanaaaduu... Xaanaaduuu, Xaanaaaduuuu, (now we are here),

In Xaaaaaanaaaaduuuuu ↑←→↑→↓←↑↑↑↓↑←↓↑↓↑↑→←→←→←→

...no fiasko... Ako vidieť, už aj zadania KáeSPéčka pohltila mánia $\mathcal{DDR}$[1]. Teraz pred i33 stoja deti (študenti, cvičiaci...) zo všetkých kútov matfyzu (a keby len z neho!) a čakajú, kedy sa budú môcť zahrať. Nooo-ho-ho, počkať počkať; v i33 musí byť predsa poriadok! A tak bolo umba-umba a bol poriadok. Chodí sa pekne po jednom ↑←↓←→[2] a vždy, keď

[1] pre šťastne nevedomých, $\mathcal{DDR}$= Dance Dance Revolution je v KSPáckej miestnosti namotávka #1. Čumí sa pri tom na blikajúce šípky a tancuje sa tak, aby bola v správnom čase nejaká (ľubovoľná) noha na správnom mieste (podľa šípky).

[2] Pozor, ľudia, ktorí hrávajú $\mathcal{DDR}$ občas vidia šípky aj na miestach, kde žiadne nie sú!

niekto ide tancovať, musí sa pekne zapísať – to aby sa vedelo a aby sa nám niekto len tak neprešmrdol.

Nuž a z celej tej eufórie z toho, že všetko pekne funguje, máme v tom poriadok a nikto sa nám neprešmŕda, po pár dňoch ti nám skrsla myšlienka: potrebujeme štatistiku. Veru, štatistika←→ je veľmi dôležitá, aby sme vedeli čo-a-ako. Chceli by sme vedieť, koľko sa nám tu toho vôbec prestrieda.... A tak sa chvíľu polenilo a potom sa nelenilo a mladá KSPač sa dala do roboty. Tu sa ešte stále v eufórií z poriadku zistil ohromný, ale ohromne pekelný bord...ehm, neporiadok. Len si to predstavte, každý sa ↓←↑↑↓→← zapisoval, kam sa mu len uráčilo. (A tak bola, mimochodom, druhá umba-umba, ale o-tom-po-tom). *Snáď najväčší problém tvorí fakt, že niekedy sa prestriedalo v rámci jednej minúty aj niekoľko ľudí a teda majú rovnaké časy; vieme povedať, že počas tej minúty tam boli, ale nevieme, ktorú časť minúty.*

No ale že sme KSPač pilná, máme to už pekne potriedené – pre každého ľudka celý utriedený zoznam časov, kedy tancoval. No a tu sa STL končí a čo s tým? No... ehm... zhruba podľa ľudovky „starý MišoF nie je doma, a mládež s tým nepohne; zavoláme riešiteľov...“ – tak si pripravte dáke kladivo...

Úloha: Daný je počet $\mathcal{DDR}$maniakov a následne pre každého týpka zoznam časov, kedy tancoval, ukončený nulou (zoznam je utriedený vzostupne). Môžete predpokladať, že časy sú celé čísla od 1 do 20000 a počet $\mathcal{DDR}$maniakov je tak do 50. Povedať presne, koľkokrát sa týpci prestriedali, nemusí byť možné. Vašou úlohou bude preto nájsť, koľko *najmenej* striedaní sa určite muselo uskutočniť.

Príklad:

Vstup:

$N = 2$
1 2 0
2 0

$N = 2$
3 5 7 7 8 0
4 6 7 7 12 0

Výstup:

1 (Označme si týpkov A a B. V prvej a na začiatku druhej bol A, potom B. Mohlo sa stať aj to, že najprv bol A, potom B a na konci druhej minúty opäť A, chceme však najmenší počet striedaní.)

5 (Zjavne v 3. minúte tancoval A, vo 4. ho vymenil B, v 5. opäť A, v 6. opäť prišiel B. Teraz vidíme, že nie je jasné, ako sa striedali počas 7. minúty. Najmenej striedaní by bolo v prípade, ak B tancoval po 6. minúte aj na začiatku 7. minúty, odtancoval si obe pesničky a potom pustil A. Ten tancoval až do konca 8. minúty, pustil B a šiel domov. Dokopy máme 5 striedaní.)

3. O maximalizácii radov

15 bodov

Na našej škole majú mnoho oddelení, kde študenti môžu vybavovať rôzne byrokratické záležitosti. Jedno z takých oddelení je aj Oddelenie Dlhých Radov (ODR).

Je to jedno z najdôležitejších oddelení. Vlastne čokoľvek si chce študent vybaviť, musí prejsť týmto oddelením. Hlavná náplň práce tohto oddelenia je vytvorenie čo najdlhšieho radu študentov čakajúcich na vybavenie. To je veľmi ťažké, hlavne keď toto oddelenie má viac kancelárií a navyše veci, ktoré treba vybaviť, sú veľmi jednoduché. Väčšinou ide len o podpísanie nejakého papiera. Zamestnanci ODR sú ale veľmi šikovní a našli veľmi efektívny spôsob ako si svoju prácu plniť čo najlepšie. Zaviedli dômyselný systém prestávok, počas ktorých neúradujú. Napríklad obedná prestávka, desiatová prestávka, kávičková prestávka I, kávičková prestávka II a mnohé iné. Samozrejme, zamestnanci si môžu podľa potreby nejakú prestávku pridať alebo ju posunúť. Udržať takýto systém v prevádzke ale vôbec nie je jedno-

duché. Každú prestávku treba stihnúť presne načas, inak by sa mohlo stať, že treba obslúžiť ďalšieho študenta a tým skrátiť rad, čo vôbec nie je dobré. Potrebovali by program, ktorý ich upozorní na začiatok každej prestávky. No a keďže oni s počítačmi nevedia robiť (ešte to tak, aby vedeli, to by sa zrýchlilo vybavovanie a skrátili rady) potrebujú vašu pomoc.

Úloha: Napíšte program, ktorý potrebujú. Váš program bude interaktívny a s okolím bude komunikovať takto: Bude prijímať správy:

- `TICK` ... túto správu dostane každú sekundu.
- `SET <d>` ... naplánuje začiatok prestávky o d sekúnd. Po prijatí tejto správy vráti nejaký identifikátor tejto prestávky.
- `DEL <id>` ... zruší naplánovanie prestávky s identifikátorom `id`.

A bude vysielať správu

- `BZZZ <id>` na začiatku prestávky s identifikátorom `id`.

Môžete predpokladať, že naraz bude naplánovaných najviac N prestávok a budú naplánované najviac M sekúnd dopredu (kde N a M sú dané na vstupe). Váš program by po spustení mal bežať večne, teda napríklad ani po veľmi dlhom čase by v ňom nemalo nič pretiecť.

Príklad:

Takto by mohla vyzerať interakcia s vašim programom.[3] ($N = 17$, $M = 47$)

```
> TICK
> TICK
> SET 2
47ksp
> TICK
> SET 4
74obed
> TICK
BZZZ 47ksp
> DEL 74obed
```

```
> SET 3
17idemejest
> TICK
> SET 2
147teraz
> TICK
> TICK
BZZZ 17idemejest
BZZZ 147teraz
```

4. O reforme ciest do práce

15 bodov

„To je škandál!“
„Už som zažila všeličo, ale toto tu ešte nebolo!“
„Ako keby nestačilo, že minulý rok zaviedli daň zo svine!“
„Hovoríš mi z duše drahá.“
„A čo sa vlastne stalo?“
„Ále, vy ste to ešte nepočuli? Veď sú toho plné správy.“
„Nepočula, práve som sa vrátila z dovolenky.“
„Nebudem chodiť okolo horúcej kaše a poviem vám to rovno. Minister chodenia do práce vydal nehorázne nariadenie.“
„...no a ja neviem či sú blbí alebo čo, ale ja proste nemám ako to spraviť, no neviem...“
„...neskáčte mi do reči drahá! Takže, ževraj od prvého mája nesmie človek cestou z práce použiť žiadnu ulicu, ktorou šiel cestou do práce, vraj aby sa nenudil.“
„No a viete, zlatko, vďaka tým žgrlošom na ministerstve ciest do prace[4] nepoznám jedinú dušu, ktorá by toto mohla spraviť, tak málo ulíc máme!“
„A vrchol všetkého je, že minister financii teraz vyhlásil, že sa dostavajú nové ulice, ale musí ich byť čo najmenej!“

[3] Čítaj najskôr ľavý, potom pravý stĺpec, nie striedavo!

[4] Nepliesť si s ministerstvom chodenia do práce! Ministerstvo chodenia **z** práce zatiaľ ešte neexistuje, ale určite chápete, k čomu táto reforma smeruje, však že?

„Presne tak. No povedzte, je normálny?“

Úloha: Máte zadané n – počet významných lokalít v meste. Momentálne sú tieto lokality poprepájané ulicami veľmi skromne, pre každé dve lokality existuje práve jeden spôsob, ako sa z jednej dostať po uliciach do druhej. Na vstupe je zadaných $n-1$ dvojíc[5] čísel lokalít, ktoré sú spojené ulicou.

Úlohou ministerstva ciest do práce (a aj vašou) je pospájať niekoľko dvojíc lokalít novými cestami tak, aby už nik nemohol mať problémy s plnením nového nariadenia. Pre každú dvojicu lokalít teda budú musieť existovať dve cesty, ktoré nemajú spoločnú ani jednu ulicu.

Môžete predpokladať, že pracovníci ministerstva sú šikovní a zvládnu postaviť novú ulicu medzi ľubovoľnou dvojicou lokalít. (Použijú nadjazdy, mimoúrovňové križovatky a podobné fičúrie.)

Výstupom programu by mal byť najmenší počet ulíc, ktoré stačí postaviť. Nezabudnite na dôkaz, že váš program naozaj nájde najmenšie možné riešenie!

Vstup:	**Výstup:**
6	2
1 3	*(Napríklad postavíme ulice 1-6 a 2-5.)*
3 4	
4 5	
2 3	
4 6	

5. Ostávame pri kryptológii

15 bodov

Aj v štvrtej časti nášho malého kryptologického seriálu si vyberieme jednu zo zaujímavých oblastí. Minule sme si niečo povedali o šifrovaní. Okrem iného sme si ukázali šifru, ktorá síce sama o sebe bola absolútne bezpečná... ale spôsoby, ktorými sme ju použili, už také skvelé neboli. (Tí z vás, ktorí vyriešili podúlohy c,d a e z minulej série, prípadne čítali jej vzorové riešenie, už vedia svoje.)

A práve o tom si budeme rozprávať dnes. Kryptológia nie je totiž len o tom vymyslieť čo najsilnejšiu a najťažšie prelomiteľnú šifru. Treba tieto šifry aj vedieť správnym spôsobom použiť. Tieto spôsoby použitia budeme volať *kryptografické protokoly*. Taký protokol nie je vlastne nič iné, ako dohodnutý postup, kto má kedy komu poslať akú správu.

Zoznámme sa najskôr s tradičným kryptografickým osadenstvom. Na úvod sú tu vaši starí známi, Alica a Bob, ktorí si chcú poslať správu. Okrem nich sú tu ale aj Eva (eavesdropper, čiže odpočúva komunikáciu) a Oskar (opponent, nepriateľ, ktorý môže do komunikácie aj aktívne zasahovať). Občas sa ukáže aj Trudy (trusted authority, niekto, komu Alica aj Bob dôverujú). V kryptografii sa používajú aj ďalšie postavy, v našom príbehu si ale vystačíme s týmito. Občas budeme namiesto mena človeka písať len jeho prvé písmenko.

Príklad protokolu z minulej série:

1. $A \to B : M \oplus K_A$
2. $B \to A : M \oplus K_A \oplus K_B$
3. $A \to B : M \oplus K_B$

Už sme videli, že tento protokol nie je odolný ani voči Eve – tej stačí vypočuť si komunikáciu a vie z nej určiť prenášanú správu M. Taký Oskar, ten dokáže napáchať ešte väčšie škody. Vo vzorových riešeniach tretej série ste sa mohli dočítať viac.

Povedzme si teraz niečo o nástrojoch, ktoré nám kryptológia ponúka pri navrhovaní protokolov. V prvom rade je to šifrovanie. Správu s zašifrovanú kľúčom K budeme značiť $\{s\}_K$. Predpokladáme, že účastníci komunikácie, ktorí K nepoznajú, nevedia efektívne z $\{s\}_K$ určiť s ani K.

[5] Ak vám nie je jasné, prečo $n-1$, radšej si to rozmyslite!

V druhom rade to sú časové pečiatky. Pre naše potreby časová pečiatka bude reťazec, v ktorom je napísaný aktuálny čas a dátum. Časové pečiatky budeme značiť T_*.

Občas sa hodí, aby niektorý účastník komunikácie poslal náhodný reťazec N_* (odborne zvaný *nonce*). Druhý účastník mu ho bude väčšinou musieť poslať späť, aby potvrdil, že jeho správa je aktuálna. Načo to bude dobré, to ešte uvidíte.

Ale dosť teórie, poďme sa pozrieť na príklad.

Protokol „žaba so širokou hubou“:
Pripomeňme si, že T (Trudy) je účastník komunikácie, ktorému Alica aj Bob dôverujú. Nech A a T majú dohodnutý kľúč K_A, nech B a T majú kľúč K_B.

1. $A \to T : A, \{B, K\}_{K_A}$
2. $T \to B : \{A, K\}_{K_B}$

Čo tento protokol hovorí? Keď chce Alica niekedy hovoriť súkromne s Bobom, vygeneruje si kľúč K, ktorý potom ona a Bob použijú. Tento kľúč K chce teraz pomocou Trudy oznámiť Bobovi. Tak pošle Trudy správu: Ja, A, chcem hovoriť s B, pošli mu kľúč K. Keďže časť správy je zašifrovaná kľúčom K_A, Trudy má istotu, že správa naozaj prišla od A. Tak pošle B správu, kde oznámi: A chce s tebou komunikovať pomocou kľúča K. Keď B dostane túto správu, vie, že je od Trudy, lebo je zašifrovaná kľúčom K_B, ktorý nik iný nepozná.

Všetko vyzerá byť v poriadku. Čo sa ale stalo, keď Alica s Bobom skúsili použiť tento protokol? Oskar si uložil na disk všetky posielané správy. Niekedy dlho po tom, ako komunikácia skončila, získal kľúč K. (Buď priamo z počítača Alice/Boba, alebo trebárs hrubou silou z nimi posielaných správ.) Teraz ale Oskar prikročí k svojmu diabolskému plánu: Tvániac sa ako Alica pošle Trudy nasledovnú správu:

1′. $O(A) \to T : A, \{B, K\}_{K_A}$

V zápise protokolu značenie $O(A)$ znamená, že je to Oskar, tváriaci sa, že je Alica.[6]

Oskar síce nevie kľúč K_A, ale správu $\{B, K\}_{K_A}$ má dávno uloženú na disku. Nič netušiaca Trudy si overí, že to šifrovala Alica a oznámi Bobovi, že Alica s ním chce hovoriť. No a nič netušiaci Bob sa začne s Alicou rozprávať, pričom použije kľúč K, ktorý mu Trudy poslala – ale ktorý už Oskar pozná! A ten jednoducho všetky Alici určené správy odchytí a tvári sa, že je ona.

Keď im toto Oskar vyviedol, videli Alica a Bob, že je zle. I upravili protokol tak, aby používal časové pečiatky. Alica si vygeneruje a v správe pošle časovú pečiatku T_A, aby ukázala, že správa je „čerstvá“. Trudy si pri prijatí správy overí, či je či je správa „dostatočne čerstvá“, ak nie, ignoruje ju. Podobne sa overí aj čerstvosť správ medzi Trudy a Bobom. Upravený protokol vyzeral takto:

Upravený protokol „žaba so širokou hubou“:

1. $A \to T : A, \{T_A, B, K\}_{K_A}$
2. $T \to B : \{T_T, A, K\}_{K_B}$

Čo ale Oskar nevymyslel? Podobne ako v prvom prípade, aj tu odchytil obe správy. Teraz mu ale nestačilo uložiť si ich, lebo by mu časom prestali platiť. Namiesto toho poslal Trudy správu:

1′. $O(B) \to T : B, \{T_T, A, K\}_{K_B}$

Pre Trudy toto vyzerá ako správa od Boba, ktorý chce komunikovať s Alicou. Časová pečiatka je čerstvá. Nuž, Trudy pošle Alici správu:

2′. $T \to O(A) : \{T'_T, B, K\}_{K_A}$

Samozrejme, Oskar si ju opäť odchytí. Čo získal? Novšiu časovú pečiatku! No a nič mu nebráni opakovať tento postup až do okamihu, kým sa nedozvie kľúč K... A potom aktuálnu správu, stále s platnou časovou pečiatkou, rovnako ako predtým použije na oklamanie Alice alebo Boba.

[6] Napr. sa pripojil na jej telefónnu linku, prípadne poslal mail, ktorý sa tvári, že je z Alicinej adresy.

Úloha:

a) Predpokladajme, že na komunikáciu je použité asymetrické šifrovanie.[7] To pre nás znamená, že hocikto vie vyrobiť šifrovanú správu určenú človeku C, ale len C si vie takúto správu prečítať.

Použitý protokol funguje tak, že A aj B si vygenerujú náhodné reťazce N_A, resp. N_B, navzájom si ich oznámia, vypočítajú si z nich kľúč $K = f(N_A, N_B)$ (f je nejaká známa funkcia) a pomocou toho budú ďalej komunikovať.

1. $A \to B : \{N_A, A\}_{K_B}$
2. $B \to A : \{N_A, N_B\}_{K_A}$
3. $A \to B : \{N_B\}_{K_B}$

Vysvetlenie: Alica pošle svoj reťazec a svoje meno, zašifrované tak, aby si to vedel prečítať len Bob. Bob si správu dešifruje, vyrobí správu určenú pre Alicu, v tej jej oznámi N_A a svoje N_B. Alica si overí, že N_A je to, čo poslala (a teda vie, že správa od Boba je čerstvá). Ak áno, oznámi mu N_B. V tomto okamihu aj Alica, aj Bob poznajú N_A aj N_B a spočítajú si kľúč K. Bob po prijatí N_B môže začať komunikáciu použitím kľúča K.

Tento protokol samozrejme môže použiť na komunikáciu ľubovoľná dvojica ľudí, nie len Alica s Bobom `:-)`.

Alica práve chce komunikovať s Oskarom a poslala mu prvú správu:

1. $A \to O : \{N_A, A\}_{K_O}$

Ukážte, ako Oskar oklame Boba – Bob si bude myslieť, že si dohodol kľúč s Alicou, v skutočnosti ho ale bude vedieť Oskar.

Hint: Paralelne s tým, ako komunikuje s Alicou, začne Oskar aj nový rozhovor s Bobom, použitím tohto istého protokolu... a podobných správ `:-)`.

b) Vráťme sa späť do situácie, kedy máme aj Trudy, ktorej Alica aj Bob veria a majú s ňou dohodnuté šifrovacie kľúče ako v príkladoch. Tentokrát si Alica s Bobom vymysleli nasledujúci protokol:

1. $A \to B : A, N_A$
2. $B \to T : B, N_B, \{A, N_A\}_{K_B}$
3. $T \to A : N_B, \{B, K, N_A\}_{K_A}, \{A, K, N_B\}_{K_B}$
4. $A \to B : \{A, K, N_B\}_{K_B}, \{N_B\}_K$

Vysvetlenie: Alica pošle Bobovi správu, že s ním chce komunikovať a náhodný reťazec N_A. Bob ak súhlasí, pošle Trudy potrebné informácie, časť z nich zašifruje, aby Trudy vedela, že sú od neho. Trudy si overí, že informácia pochádza od Boba – a keďže A bolo zašifrované, vie, že Bob chce naozaj hovoriť s Alicou. Trudy im vygeneruje kľúč K. Alici pošle Bobovo náhodné slovo N_B, správu pre ňu a správu pre Boba. Alica si odšifruje správu určenú pre ňu, overí si N_A (teda vie, že jej prvá správa prešla cez Boba) a dozvie sa kľúč K. Pošle Bobovi jeho správu. Navyše mu oznámi, že ona úspešne získala kľúč K – tak, že mu pošle $\{N_B\}_K$. Ak aj Bobovi všetko sedí, vie, že si práve s Alicou dohodol kľúč K.

Aspoň tak si to Alica s Bobom predstavovali. Oskar však opäť niečo vyhútal. Dokáže Alicu presvedčiť, že si dohodla kľúč s Bobom, hoci to nebude pravda.

Hint 1: Oskar sa nedozvie K.

Hint 2: Začiatok útoku bude vyzerať takto:

1. $A \to E(B) : A, N_A$
1′. $E(B) \to A : B, N_A$
2′. $A \to E(T) : A, N'_A, \{B, N_A\}_{K_A}$
...

[7] Príkladom je šifra RSA, viď `http://mathworld.wolfram.com/RSAEncryption.html`