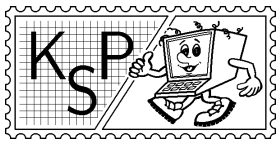

Korepondenčný seminár z programovania
XIX. ročník, 2001/2002

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

KSP finančne podporuje nadácia Open Society Fund Bratislava a IUVENTA – zariadenie pre vožný čas detí, mláde« e a dospelých MICROSTEP spol s.r.o.

Kategória Z

Milí rieitelia,

jesenné sústredenie máme úspene za sebou a vám sa (konečne?) dostávajú do rúk vzorové rieenia prvého kola. Tak si ich pozorne prečítajte, aby ste vedeli, ako sa tie úlohy mali riei ̧. A ak sa dočítate a« na koniec, nájdete tam... prekvapenie. (psst... je tam samozrejme výsledková listina... ale najskôr povinne prečíta ̧ vzorové rieenia!)

Po pou«ití záchodu si umyte ruky.

KSPáci

1. Zlatá retiazka

opravoval WSX
(max. 15 bodov)

Keď Ivanhoe dostal vae rieenia, hneď sa chcel pusti ̧ do pižovania očiek. Keď« e poklad bol tak dobre chránený, bol aj dostatočne vežký. Rýchlo sa pustil do zadávania vstupných dát a podarilo sa mu nájs ̧ len 2 rieenia, ktoré stihli da ̧ správne rieenie tak rýchlo, aby ete stihol utiec ̧ drakovi zvanému Mandrake, ktorý je vežmi rýchly.

Vae rieenia mo«no rozdeli ̧ na niekožko skupín podža ich časovej zlo«itosti, podža ktorej boli bodované:

- Lineárne (od vežkosti vstupu) $O(N+M)$: Rieenia, ktoré pou«ívali modifikovaný Dijkstrov algoritmus alebo frontu. Za takéto rieenie sa dalo získa ̧ **15 bodov**.
- $O(N^2)$: Rieenia, ktoré pou«ívali Dijkstrov algoritmus. Bodová hranica **14 bodov**.
- $O(NM)$: Najpočetnejia skupina rieení zalo«ených na mylienke pre ka«dú hĺbku a vetky vrcholy predchádzajúcej hĺbky prejs ̧ vetky hrany a tým zisti ̧ ďaliu hĺbku. Max **12 bodov**.
- $O(N^3)$: Troku horie rieenie ako $O(NM)$, hlavne pre riedke grafy, **11 bodov**.
- iné pomalie polynomiálne rieenie: Dalo sa získa ̧ **10 bodov**.
- backtrack: **9 bodov**.
- nefungujúce rieenia: Maximálne **3 body**.

Ďalie body sa dali strati ̧ za chýbajúci alebo nedostatočný slovný popis algoritmu, zdôvodnenie správnosti, odhad časovej zlo«itosti a za eleganciu rieenia.

Ako teda napísa ̧ lineárne rieenie? Ako vetci správne zistili, treba nájs ̧ najkratiu cestu medzi očkami 1 a 2 a Ivanhoe si mô« e odpíli ̧ očká, ktoré nepatria do tejto cesty. Prečo to tak funguje? Zdôvodnenie je jednoduché – čím menej očiek bude v spojení medzi 1 a 2, tým viac sa dá zobra ̧. Najmenej ich bude práve v najkratej ceste. Teraz treba prís ̧ na to, ako nájs ̧ túto cestu. Vzorové rieenie pou«íva frontu, v ktorej sa nachádzajú u« spracované očká. Fronta je dátová truktúra, do ktorej vieme vklada ̧ údaje a vybera ̧ ich z nej. Pritom funguje tak, « e najskôr bude vybratý údaj, ktorý sme do nej najskôr vlo«ili. Svoj názov získala fronta (alebo tie« rad) podža radu žudí čakajúcich na obed. Tam tie« [1] kto prv príde, ten prv dostane jes ̧. Jeden spôsob, ako takú frontu napísa ̧, je ma ̧ prvky ulo«ené v nejakom súvislom kuse

[1] viac-menej

poža a pamäta¸ si, kde začína a kde končí. Na jednom konci vkladáme nové očká a na druhom vyberáme tie najskôr vlo«ené. Na začiatku dáme do fronty len prvé očko. Ďalie očká budeme pridáva¸ nasledovným spôsobom: Vyberieme z fronty jedno očko, nájdeme vetky také hrany (spojenia očiek), ktoré spájajú toto očko s očkom, ktoré ete nebolo spracované a toto nespracované očko pridáme na druhý koniec fronty. Kým budeme pokračova¸ ďalej, treba si objasni¸ nejasné veci – ako budeme hžada¸ hrany a ako budeme zis¸ova¸, či bolo očko u« spracované. Aby sme si nepokazili časovú zlo«itos¸, bolo by dobré, keby sme si hrany pamätali u« v takej forme, «e pre ka«dý vrchol (očko) by sme mali zoznam vrcholov, ktoré sú s ním spojené. Ka«dú hranu si teda budeme pamäta¸ dvakrát. Raz ako druhý vrchol v zozname hrán prvého vrcholu a druhý krát presne naopak. Ako rýchlo zisti¸, či bol vrchol spracovaný? Tak, «e si spravíme pole, v ktorom si budeme pamäta¸ pre ka«dý vrchol, či u« bol spracovaný. V«dy, keď spracujeme nejaký vrchol (pridáme ho do fronty), nastavíme tomuto vrcholu, «e u« bol spracovaný.

Ete stále vak nevieme, čo má táto fronta spoločné s hžadaním najkratej cesty. Nech *hĺbka* vrcholu vyjadruje dĺ«ku jeho najkratej cesty k prvému vrcholu. Potom na začiatku máme vo fronte prvý vrchol s hĺbkou 0. V prvom kroku tento vrchol vyberieme z fronty a pridáme do nej niekožko vrcholov s hĺbkou 1. Teraz budeme vybera¸ vrcholy s hĺbkou 1 a vkladané budú ma¸ hĺbku o jedna vyiu. To zdôvodníme tak, «e keby mali ni«iu hĺbku, tak by sme ich u« spracovali predtým a vyiu nemô«u ma¸, lebo určite existuje cesta s touto dĺ«kou. Takto pokračujeme a v«dy vyberáme vrcholy s hĺbkou i a vkladáme s hĺbkou $i+1$. Túto čas¸ programu preruíme buď vtedy, keď sme vybrali vetky vrcholy, alebo keď sme vlo«ili vrchol 2 a tým sme nali najkratiu cestu k nemu.

Teraz by sme vak chceli vedie¸ celú cestu. Na to si spravíme ďalie pole, v ktorom si budeme pamäta¸ vrchol, cez ktorý sme sa dostali k danému vrcholu. Pri pridávaní do fronty vieme, cez ktorý vrchol sme sa tam dostali, či«e si to rovno zapíeme do tohto poža. Aby sme zistili celú cestu, pôjdeme od vrcholu 2 po predchodcoch, a« kým nenarazíme na vrchol 1. Tieto vrcholy si nejako vhodne označkujeme v nejakom poli a potom u« len vypíeme vrcholy, ktoré sa tu nie sú označkované, a teda ich mô«eme odstráni¸.

Dôkaz správnosti: Na začiatku sme si dokázali, «e ak vypíeme prvky nepatriace do najkratej cesty medzi vrcholmi 1 a 2, je to rieením úlohy. Dokázali sme si aj to, «e ak pou«ijeme frontu daným spôsobom, nájdeme tým najkratiu cestu. Spojením týchto dvoch implikácií dostaneme, «e ak pou«ijeme tento algoritmus, dostaneme rieenie úlohy. □

Odhad zlo«itosti: Pamä¸ová zlo«itos¸ algoritmu je $O(N+M)$ – pre ka«dú hranu a pre ka«dý vrchol si pamätáme len kontantné mno«stvo údajov. Časová zlo«itos¸ je tie« $O(N+M)$. Do fronty vkladáme maximálne N krát, či«e aj vyberáme maximálne N krát. Pre ka«dý vrchol (keď ho vyberieme z fronty) ideme v cykle maximálne jeden krát po „jeho“ hranách, teda ka«dú z M hrán navtívime kontantný počet krát. Zistenie spiatočnej cesty je v čase $O(N)$, to nám časovú zlo«itos¸ nepokazí.

Listing programu:

```
Program Zlata_retiazka;
Const MAXN=100;
Var N:Integer;
    Hrany:Array[1..MAXN, 1..MAXN] Of Integer;
    PHran:Array[1..MAXN] Of Integer; { stupen vrcholu }
    Bol:Array[1..MAXN] Of Byte;      { 0=nebol, 1=bol, 2=najlepsia cesta }
    Spat:Array[1..MAXN] Of Integer;  { cesta spat }

Procedure Nacitaj;                   { nacitanie vstupu }
Var I, J, M:Integer;
Begin
```

```
  ReadLn(N, M);
  For I:=1 To N Do
  Begin
    PHran[I]:=0;
    Bol[I]:=0;
  End;
  While M>0 Do
  Begin
    ReadLn(I, J);
    Inc(PHran[I]);
    Inc(PHran[J]);
    Hrany[I, PHran[I]]:=J;
    Hrany[J, PHran[J]]:=I;
    Dec(M);
  End;
End;

Procedure ZistiCestu(Zkade, Kam:Integer);
Var Queue:Array[1..MAXN] Of Integer; { fronta }
    QR, QW:Integer;                  { zaciatok a koniec fronty }
    I:Integer;
Begin
  Queue[1]:=Zkade;
  Bol[1]:=1;
  QR:=1;
  QW:=1;
  While (QR<=QW) And (Bol[Kam]=0) Do
  Begin
    For I:=1 To PHran[Queue[QR]] Do
      If Bol[Hrany[Queue[QR], I]]=0 Then
    Begin
      Inc(QW);
      Queue[QW]:=Hrany[Queue[QR], I];
      Bol[Queue[QW]]:=1;
      Spat[Queue[QW]]:=Queue[QR];
    End;
    Inc(QR);
  End;
  If Bol[Kam]=0 Then
  Begin
    WriteLn('Nema riesenie.');
    Halt;
  End;
End;

Procedure OznacCestu(Zkade, Kam:Integer);
Begin
  Bol[Kam]:=2;
  While Kam<>Zkade Do
  Begin
    Kam:=Spat[Kam];
```

```
    Bol[Kam]:=2;
  End;
End;

Procedure Vypis;
Var I:Integer;
Begin
  WriteLn('Treba vyhodit ocka:');
  For I:=1 To N Do
    If Bol[I]<>2 Then
      WriteLn(I);
End;

Begin
  Nacitaj;
  ZistiCestu(1, 2);
  OznacCestu(1, 2);
  Vypis;
End.
```

2. Z trpaslíčej chalúpky

opravovalo YoYo δ
(max. 15 bodov)

Microlandskí trpaslíci sa s rados¸ou vrhli na vae rieenia dúfajúc, «e im pomo«u vyriei¸ zvadu. A tak namiesto tlačenia fúrika a pridŕ«ania laty pustili sa pú¸a¸ vae programy. Skoro vetky dávali správne výsledky, len na microlandské pomery boli vééézmi pomalé. Na¸astie sa ale nalo aj niekožko (aj keď naozaj len zopár) rýchlych rieení. A tak mohli trpaslíci rýchlo ukonči¸ zvadu a pusti¸ sa do stavby chalúpky.

Ako boli teda vae rieenia bodované? Hlavným kritériom bola samozrejme čitateżnos¸ a estetickos¸ rieení. Rieenia písané rukou nemali ancu dosta¸ viac ako 5 bodov. Rieenia tlačené na tlačiarni boli hodnotené v závislosti od pou«itého fontu, pekného oddelenia popisu od kódu programu od 5 do 12 bodov. Ďalie body sa dali získa¸ za syntax higlightning. Rieenia písané v TEXu získavali ďalie 2 body. Dievčatá dostávali bonus 5 bodov, plus ďalie body za pekné obrázky. Súčet ale samozrejme nemohol prevýi¸ 15 bodov...

Asi takto by mohlo vyzera¸ hodnotenie v nejakom rozprávkovom seminári. U nás vak hodnotíme podža iných pravidiel. Hlavným kritériom (samozrejme po správnosti) bola časová zlo«itos¸. Tá rozdelila vae rieenia na tri skupiny, dve vežké a jednu maličkú:

- Kvadratické $O(N^2)$: Rieenia spočívajúce v zotriedení trpaslíkov a v overení správnej podmienky. Najpočetnejia skupina, za takéto rieenie sa dalo získa¸ max **10 bodov**.
- $O(N \log N)$: Menej početná skupina rieení, ideovo nie nepodobná predchádzajúcej, len pou«ívajúca rýchlejie triedenie (väčinou QuickSort). Bodová hranica **12 bodov**.
- Lineárne $O(N)$: Skupina tvorená zopár[2] rieeniami, ideovo nie nepodobnými vzorovému rieeniu. Bohu«iaž sa tu ale nalo niekožko chybných. Tu sa dalo získa¸ vytú«ených **15 bodov**.

Samozrejme, ako zvyčajne sa nemilosrdne stŕhali body za chýbajúci slovný popis rieenia, prípadne za iné drobné chybičky. Z tých najčastejích sa patrí spomenú¸ delenie nulou v prípade vstupu s jedným trpaslíkom ($N = 1 \Rightarrow N - 1 = 0$), alebo pristupovanie do poža mimo jeho rozsah.

[2] asi 5

Aké je teda lineárne rieenie? Jednoduché. Ako väčina vzorových rieení. Len bolo treba neda˛ sa oklama˛ zadaním. Ak zadanie hovorí: „ak sa trpaslíci usporiadajú“, neznamená to, «e musíme hňed triedi˛. Netreba trpaslíkov niekam umiestňova˛ a zis˛ova˛, či takéto umiestnenie vyhovuje. Dá sa na to ís˛ aj z opačnej strany. Skúsme pre ka«dáho trpaslíka zisti˛, kde by mal by˛ postavený.

Urobme na zemi N čiarok v pravidelných rozostupoch. Na prvú postavme najmenieho trpaslíka, na poslednú najväčieho. A nech nám teraz niekto posiela ostatných v žubovožnom poradí. Keď zmeriame dotyčnému výku, vieme presne poveda˛, kam sa má postavi˛, aby latu podopieral[3] (treba si uvedomi˛, «e to bude niekde medzi najmenším a najväčím trpaslíkom). Ak zistíme, «e sa má postavi˛ na nejakú čiarku, tak je to v poriadku. Ak by mal stá˛ niekde medzi, tak máme problém. Teda presnejie trpaslíci majú problém – niekto sa bude ulieva˛. Ete sa mô«e sta˛, «e by sme ho mali postavi˛ na čiarku, kde u« nejaký trpaslík stojí. V tom prípade sa tie« nemusíme ďalej trápi˛, trpaslíci sa budú musie˛ poháda˛.

U« vieme ako na to, preberme si to teda trochu maticky. Ak majú by˛ hlavy trpaslíkov po usporiadaní na jednej priamke, musia ich výky tvori˛ aritmetickú postupnos˛. To je pozorovanie, ktoré neuniklo väčine rieitežov. Venujme sa ale radej predstave laty ako priamky „spájajúcej“ hlavy trpaslíkov. To nám umo«ní jednoducho vyráta˛ pre danú výku, kam patrí. Označme si v_i výku i-teho trpaslíka a min (max) nech je číslo najmenieho(najväčieho) trpaslíka. Priamka – lata prechádza bodmi $[1, v_{min}]$ a $[N, v_{max}]$ (aj prvý aj posledný trpaslík ju musí podopiera˛). Nech teraz priiel trpaslík s výkou y. Kam ho postavi˛? Na tožkú čiarku, aby výka laty nad ňou bola práve y. Pre výku laty nad bodom x platí:

$$y = (x-1) \cdot \frac{v_{max} - v_{min}}{N-1} + v_{min}$$

Označme $d = \frac{v_{max} - v_{min}}{N-1}$, po upravení dostávame:

$$x = 1 + \frac{(y - v_{min})}{d}$$

Nesmieme ale zabudnú˛ na malé drobnosti ako $N - 1 \neq 0$ a $v_{max} - v_{min} \neq 0$.

Pre toto x potrebujeme iba zisti˛, či je celočíselné. Ak je, treba ete overi˛, či u« také nebolo a ak nie, tak si ho samozrejme nesmieme zabudnú˛ poznači˛. O to sa vo vzorovom programe stará pole `bol`. Ak tento test zbehne pre vetkých trpaslíkov, budú ma˛ hlavy na jednej late/priamke, ináč nie.

Odhad zlo«itosti: Ako u« bolo povedané, časová aj pamä˛ová zlo«itos˛ je lineárna. Načítanie je očividne lineárne a nájdenie v_{max} a v_{min} tie«. Test jedného trpaslíka je kontantný a potrebujeme otestova˛ lineárne veža trpaslíkov. Časová zlo«itos˛ celého algoritmu je teda $O(N)$. Pamäta˛ si potrebujeme len výky N trpaslíkov, čo tie« znamená lineárnu pamä˛ovú zlo«itos˛. Tak a hor sa k fúrikom...

Listing programu:

```
Program Z_trpaslicej_chalupky;
const MaxT=100;
var N:integer;                                          {počet trpaslíkov}
    T:array[1..MaxT] of real;                    {pole s výkami trpaslíkov}
    bol:array[1..MaxT] of boolean;{pole označujúce, ktoré pozície u« boli}
    i,pocet:integer;
    min,max,d,poz:real;
```

[3] v praxi ho stačí postavi˛ k tomu najvyiemu a tlači˛ smerom k najni«iemu, a« kým hlavou nenarazí o latu

```
begin
  readln(N);                                          {počet trpaslíkov}
  for i:=1 to N do readln(T[i]);                    {výka i-teho trpaslíka}
  if N=1 then begin                                {ak je iba jeden trpaslík}
    Writeln('Ano');                                {tak nie je o čom        }
    exit;
  end;
  for i:=1 to N do bol[i]:=false;
  min:=T[1]; max:=T[1];
  for i:=2 to N do                                  {nájdeme najmenieho    }
    if T[i]>max then max:=T[i] else                 {a najväčieho trpaslíka}
      if T[i]<min then min:=T[i];
  d:=(max-min)/(N-1);                             {rozdiel medzi nasledujúcimi}
  pocet:=0;
  i:=1;
  repeat
    if d=0 then begin
      if T[i]=max then poz:=i else poz:=i+0.5;
    end else
      poz:=1+(T[i]-min)/d;                  {pozícia i-teho trpaslika v rade}
    if (frac(poz)=0) and (not bol[Trunc(poz)]) then begin
      bol[Trunc(poz)]:=true;  {označíme si, «e toto miesto je u« obsadené}
      inc(pocet);                          {zvýime počet správne zaradených}
    end else
      pocet:=-1;

    inc(i);
  until (i>N) or (pocet=-1);

  {ak sme zaradili vetkých trpaslíkov, tak mô«eme polo«i˛ latu...}
  if pocet=N then Writeln('Ano') else Writeln('Nie');
end.
```

3. Z tajného laboratória RSA

opravoval Braňo
(max. 15 bodov)

Chvížku (naozaj iba maličkú) poďme rozmýža˛. Sžubujem, «e to bude bolie˛ len vežmi máličko.[4]

Kedy číslo N, zapísané v sústave so základom K, končí i nulami?

$$N = a_k K^k + a_{k-1}K^{k-1} + \ldots + a_{i+1}K^{i+1} + a_i K^i + 0K^{i-1} \ldots + 0$$
$$N = K^i(a_k K^{k-i} + a_{k-1}K^{k-i-1} + \ldots + a_{i+1}K + a_i)$$

No predsa práve vtedy, keď je deliteľné číslom K^i. Keď nájdeme deliteža K^i čísla N, ktorý má najväčie i, tak sme úlohu (skoro) vyrieili. Pri základe K bude ma˛ toti« N na konci i núl. Ak by K bolo zlo«ené a $K^i \mid$, tak je deliteľné nejakým prvočíslom p. Ale potom aj $p^i \mid N$, a teda aj p je rieením. Preto stačí uva«ova˛ prvočíselné K.

Ka«dé číslo (teda aj N) sa dá jednoznačne rozlo«i˛ na súčin prvočísel $N = p_1^{n_1} p_2^{n_2} \ldots p_l^{n_l}$. Číslo K^i musí by˛ deliteľom N a preto sa dá z neho vyňa˛. A čo mô«me vyňat z $N =$

[4] ako vieme, rozmýžanie predsa v«dy bolí

$p_1^{n_1} p_2^{n_2} \dots p_l^{n_l}$, aby bolo tvaru K^i? Preto my mô«eme pokojne ako optimálny kžúč zvoli¸ prvočíslo p_i s najväčím n_i.

Programu rieiaciemu úlohu teda stačí skúa¸ zaradom vetky prvočísla a delí¸ nimi N, pokiaž to ide. Pre ka«dé si spočítame, kožkokrát sa nám to podarilo. Keď z N ostane u« iba 1, tak skončíme a vypíeme najlepie.

Ono ani tak nevadí, keď budeme zaradom skúa¸ vetky čísla a nielen prvočísla. Ak skúané číslo je zlo«ené, tak u« máme N predelené vetkými prvočíslami z jeho rozkladu a N ním teda nebude deliteľné. Taktie« nemusíme skúa¸ čísla väčie ako $\sqrt{N}$. Zápis N v základe N je v«dy (a« na $N = 1$) 10. Preto stačí hžada¸ kžúče so silou aspoň 2, teda $K^2|N; K \leq \sqrt{N}$

Odhad zlo«itosti: Prechádzame čísla $1 \dots \lfloor\sqrt{N}\rfloor$, to je $O(\sqrt{N})$ krokov. Navye ete vo vnútornom cykle delíme, pokiaž je zvyok rovný 0. Len«e to sa stane iba tožkokrát, kožko je prvočísel v rozklade čísla N, maximálne $O(\log N)$. Celková časová zlo«itos¸ teda je $O(\sqrt{N} + \log N)$. Preto«e $\log N < \sqrt{N}$, dá sa odhad upravi¸ na tvar $O(2\sqrt{N}) = O(\sqrt{N})$. Pama¸ová zlo«itos¸ je kontantná, lebo nepou«ívame «iadne polia ani rekurziu t.j. $O(1)$.

A čo vae rieenia? Kožko bodov mohli dosta¸?

- Rieenia rýchlejie ako vzorák - viac podža rýchlosti[5]
- Rieenia rovnako rýchle ako vzorák $O(\sqrt{N})$ – **13 bodov**
- Rieenia o kúsok pomalie ako vzorák $O(N)$ – **11 bodov**
- Rieenia ete o kúsok pomalie – **7 a« 10 bodov**
- Nefunkčné rieenia – max. **3 body**
- Popis programu a elegantnos¸ kódu – mo«né **2 body** k dobru

Listing programu:

```
program Z_tajneho_laboratoria_RSA;
var N, K, maxPoc : integer;
    i, poc       : integer;
begin
  write('Zadaj N : ');
  readln(N);          {mlcky predpokladame, ze N je kladne}

  maxPoc:=1;          {Pri zaklade N mame na konci 1 nulu}
  K:=N;
  if N=1 then begin  {Ak N=1, kazdy kluc je rovnoko dobry}
    maxPoc:=0; K:=0; {lebo nikdy nebude 0 na konci}
  end;

  for i:=2 to trunc(sqrt(N)) do begin
    poc:=0;
    while N mod i = 0 do begin    {skusame delit cislom i}
      N:=N div i;                 {pokial mozme}
      inc(poc);   {spocitame kolkokrat sa nam to podarilo}
    end;
    if poc>maxPoc then begin
      K:=i; maxPoc:=poc;
    end;
  end;
```

[5] Ide to aj lepie, ale je to **o dos¸** zlo«itejie = grc

```
  writeln('Najlepsi kluc je ',K);
end.
```

opravoval Dávidko
(max. 15 bodov)

4. Zhrdzavené potrubie

Drvivá väčina z vás vyrieila túto úlohu k mojej spokojnosti. **Dva bodíky** som vak musel stŕha ̧ za zlú, t.j. a« lineárnu pamä ̧ovú zlo«itos ̧, preto«e ste načítali diery najprv do poža a pracovali ste s nimi v poli, čo je vak a« zbytočný luxus. Pár nezbedných jedincov navye triedilo diery, napriek tomu, «e v zadaní bolo jasne napísané, «e sú u« utriedené. No a triedenie, ako mo«no viete, mo«no nie, nejde spravi ̧ v lineárnom čase, tak«e to pokazí výslednú časovú zlo«itos ̧ a aj vá bodový zisk o ďažie **4 body**. Podobne dopadli aj ostatné rieenia s časovou zlo«itos ̧ou $O(N^2)$. Zlé rieenia dostali najviac **3 body**.

Za nedostatočné zdôvodnenie správnosti alebo popis som stŕhal po bode, za úplne chýbajúce 2. Za chýbajúci odhad časovej a pamä ̧ovej zlo«itosti bol ďalí bodík dole.

Vzorové rieenie: Kvôli žahiemu vyjadrovaniu si predstavme, «e potrubie ide zžava doprava. ahko si vimneme, «e najžavejiu záplatu mô«eme polo«i ̧ od najžavejej diery. Viac vpravo najžavejiu záplatu da ̧ nemô«eme, inak by sme nezakryli najžavejiu dieru. Ak by zase le«ala viac vžavo, tak posunutím doprava do vyie spomenutej polohy nič nestratíme.

Teraz u« vieme, kam mô«eme da ̧ nau najžavejiu záplatu. Tá zakryje nejaké diery. Podobnou argumentáciou zdôvodníme, «e druhú najžavejiu záplatu mô«eme da ̧ od najžavejej nezakrytej diery. Podobne s ďalími záplatami.

Ná postup si mo«no predstavi ̧ takto: Predstavme si, «e sa postavíme na začiatok potrubia a pochodujeme po ňom smerom doprava. Ak zbadáme dieru, tak od nej dáme záplatu. Ignorujeme diery, ktoré sú od poslednej polo«enej záplaty vzdialené do jedného metra, preto«e sú zaplátané naou záplatou. Toto robíme, a« kým neprídeme na koniec potrubia.

Program potom bude vyzera ̧ jednoducho. Budeme postupne načítava ̧ diery. Keď«e sú utriedené, tak ich načítavame presne v tom poradí, ako by sme ich pri naom pochode zbadali. Pamätáme si, odkiaž sme polo«ili poslednú záplatu. Pri načítaní ďalej diery len zistíme, či je poslednou záplatou zakrytá, vtedy neurobíme nič. Ak nie je zakrytá, tak polo«íme novú záplatu t.j. vypíeme jej pozíciu a zapamätáme si ju ako polohu novej poslednej záplaty.

Tento algoritmus urobí nanajvý niekožko málo operácií pre ka«dú z dier, preto povieme, «e jeho časová zlo«itos ̧ je lineárna od počtu dier.[6] Matematici tento fakt povedia takto: Časová zlo«itos ̧ je $O(N)$.

Podobne je to s pamä ̧ovou zlo«itos ̧ou. Program si pamätá len niekožko málo čísel, preto povieme, «e časová zlo«itos ̧ je kontantná.[7] Pododne ako predtým matematik povie toto: Pamä ̧ová zlo«itos ̧ je $O(1)$.

Listing programu:

```
Program Zhrdzavene_potrubie;
var N,i:integer;
    diera,zaplata:real;        { pozicia diery, lavy koniec zaplaty }
begin
  read(N);
  zaplata:=-999999;    { nieco hrozne vlavo }
  for i:=1 to N do
  begin
```

[6] Niekedy sa jednoducho povie, «e časová zlo«itos ̧ je lineárna a ikovný čitatež si má domyslie ̧ vzhžadom na aký paramenter.

[7] Niekto by opä ̧ mohol poveda ̧ *kontatná od počtu dier*. Kontanta má ale tú výhodu, «e je kontantná od žubovožného parametra.

```
    read(diera);
    if diera>zaplata+1 then  { Ak povodna zaplata uz nedociahne... }
    begin
      zaplata:=diera;                    { ...dame novu zaplatu...}
      writeln('zaplata od ',zaplata);    { ...a vypiseme ju. }
    end;
  end;
end.
```

Opravovali JanoS a Evka
(max. 11 bodov)

5. Myhill a Nerode čítajú báseň

Opravovanie tejto úlohy bolo nadmieru jednoduché, čoho sa hneď chytila Evka. Za správne rieenie ste dostali **10 bodov** a ak ste aj napísali pôvod básní dostali ste **jeden bod** navye. No a mne ostalo napísa¸ vám nejaký pekný vzorák. Ako si mnohí z vás vimli, úloha bola zaifrovaná jednoduchou *Cézarovou ifrou*, prensejie posunom o 13 písmen. Teda ka«dé písmeno bolo posunuté o 13 miest v abecede – takto sa z `A` stalo `N`, z `B` sa stalo `O`, ... Takto ste sa mohli dopracova¸ k názvom básní, ktoré sú **Taradúr** a **Jabberwocky**. Tí bystrejí prili aj na to, «e je to tá istá báseň, len Taradúr je jej slovenský preklad. Tí pátravejí sa dopátrali (najobžúbenejia pátracia metóda bola `www.google.com :-)` k tomu, «e je to báseň od *Lewisa Carrolla* a z knihy *'Through the Looking-Glass and What Alice Found There, 1872'*, ktorá bola vydaná aj po slovensky ako súčas¸ knihy *'Alica v krajine zázrakov a Za zrkadlom, Mladé Letá 1981'*. Táto skvelá kni«ka by mala by¸ povinným čítaním ka«dého programátora, tak«e ma poteilo, «e ju tožkí poznáte. Ked«e ste niektorí boli zarazení anglickým textom, v dnenom netradičnom vzoráku vám ponúkam perličku, a to uká«ky prekladu tejto básne do rôznych jazykov, aby ste videli, ako si rôzni prekladatelia poradili s týmto krásnym jazykolamom. Kto deifruje, ktoré z nich je v akom jazyku, je na najlepej ceste sta¸ sa lingvistom, tak«e veža tastia!

Tlachapoud (Alojz a Hana Skoumalovi)

Je svačvečer. Lysperní jezeleni
se vírně vrtáčejí v mokřavě.
Vetchaří hadroci jsou roztruchleni
a selvy sytí teskonoskuhravě.

Stře« se, stře« Tlachapouda, milý synu,
má tlamu zubatou a ostrý dráp.
Pták Zlokrv u« se těí na hostinu,
vzteklitě číhá na tě Pentlochňap.

vahlav (Jaroslav Císař)

Bylo sma«no, lepě svihlí tlové
se batoumali v dálnici
chrodní byli borolové
na mamné krsy «arnící.

Ó synu, stře« se vahlava,
má zuby, drápy přeostré,
stře« se i Ptáka Neklava,
zuřmíci Bodostre!

ja'pu'vawqoy (Keith Lim)
puqloDwI' ja'pu'vawq Dayep
pe'vIl chop Ho'Du'Daj; pe'vIl Suq pachDu'Daj
Ha'DIbaH puv juchyub yIyep
bInDepSuHach vaQeHmuS ghombe' DanIDjaj

'etlhDaj veSpatlh HujtaH ghopDaj
jagh HoSlaw' law' veqlargh Hos puS! nIteb nej nI'
vaj Sor tamtam, ghaH retlhDaq Qam
nI'be' leSII' ghah Sor retlhDaq 'ej ghaH QubII'

Gaberbocchus (Hassard H. Dodgson)
Hora aderat briligi. Nunc et Slythia Tova
Plurima gyrabant gymbolitare vabo;
Et Borogovorum mimzebant undique formae,
Momiferique omnes exgrabure Rathi.

Cave, Gaberbocchum moneo tibi, nate cavendum
Unguibus ille rapit. Dentibus ille necat.
Et fuge Jubbubum, quo non infestior ales,
Et Bandersnatcham, quae fremit usque, cave.

Dromeparden (Zinken Hopp)
Det lôystra. Lanke lågmælt sjor
hang darme frå det tarve lap.
So stige låg den rumse kor
i sovepaskens gap.

For Dromeparden du deg akt,
min djerve son! Med ilske klôr
fyk Starefuglen ut på jakt
i bygdene mot sôr.

De Krakelwok (Westervaarder & Kurpershoek)
't Was bradig en de slijp'le torfs
Driltolden op de wijde weep:
Misbrozig stonden borogorfs,
't Verdoolde grasvark schreep.

Mijn zoon, vrees de Krakelwok!
Zijn kakement, zijn grepe klauw!
Vrees ook de Jubjub-vlerkenbrok,
De gritse Bandjegauw!

Jerigóndor (Francisco Torres Oliver)
Cocillaba el día y las tovas agilimosas
giroscopaban y barrenaban en el larde.
Todos debirables estaban los burgovos,
y silbramaban las alecas rastas.

Cuídate, hijo mío, del Jerigóndor,
que sus dientes muerden y sus garras agarran!
¡Cuídate del pájaro Jubjub, y huye
del frumioso zumbabadanas!

Szajkóhukky (Weőres Sándor)
Volt egy brillős, a csuszbugó
Gimbelt és gált távlengibe,
Minden mimicre purrogó,
Mómája ingibe.

Vigyázz, jön Szajkóhukk, fiam!
Foga maró, karma furó!
Ügyelj, Csapcsip madár zuhan
S a brunkós Bromboló!

Dżabbersmok (Maciej Słomczynski)
Było smaszno, a jaszmije smukwijne
Świdrokrętnie na zegwniku wężały,
Peliczaple stały smutcholijne
I zbłąkinie rykoświstąkały.

Ach, Dżabbersmoka strzeż się, strzeż!
Szponów jak kły i tnących szczęk!
Drżyj, gdy nadpełga Banderzwież
Lub Dżubdżub ptakojęk.

Le Jaseroque (Frank L. Warrin)
Il brilgue: les tôves lubricilleux
Se gyrent en vrillant dans le guave.
Enmîmés sont les gougebosqueux
Et le môme rade horsgrave.

Garde-toi du Jaseroque, mon fils!
La gueule qui mord; la griffe qui prend!
Garde-toi de l'oiseau Jube, évite
Le frumieux Band-a-prend!

Der Jammerwoch (Robert Scott)
Es brillig war. Die schlichte Toven
Wirrten und wimmelten in Waben;
Und aller-mümsige Burggoven
Die mohmen Räth' ausgraben.

Bewahre doch vor Jammerwoch!
Die Zähne knirschen, Krallen kratzen!
Bewahr' vor Jubjub-Vogel, vor
Frumiösen Banderschntzchen!

Výsledková listina po 1. sérii kategórie KSP-Z

	Meno a priezvisko	kola	Trieda	11	12	13	14	15	Σ
1.	Marek Jančuka	Gym. Párovská Nitra	2	14	12	13	13	10	62
2.	Gabriel Války			11	10	15	14	11	61
3.	Ladislav Ruttkay	Gym. kolská Spi. Nová Ves	4	15	12	9	13	11	60
4.	Imrich ivčák	Gym. Komenského Trebiov	4	11	12	13	12	11	59
	Milan atka	Gym. Liptovský Hrádok	3	12	12	13	12	10	59
6.	Michal Kevický	Gym. Grösslingová BA	2	8	9	13	15	11	56
	Juraj Ladicky	Gym. Partizánske	4	9	10	15	12	10	56
8.	Martin Dobia	Gym. udovíta túra Trenčín	3	8	10	13	14	10	55
	Martin Rejda	Gym. Grösslingová BA	2	9	10	11	14	11	55
10.	Michal Dzetkulič	Gym. P. Horova Michalovce	1	11	10	10	13	10	54
11.	Daniel Vanco	Gym. Mudroňova Preov	4	6	12	10	14	11	53
12.	Tomá Práznovský	Gym. Sereď	2	9	8	12	12	11	52
13.	Jakub Závodný	Gym. Grösslingová BA	2		10	13	15	11	49
14.	Ján Lučanský	Gym. Komenského Trebiov	4		14	9	12	11	46
	ubomír Kundrák	Ev. lýceum BA	2		10	13	12	11	46
16.	Frantiek mitala	Gym. Farská Nitra	3		10	13	12	10	45
	Ondrej Hiriak	Gym. Mudroňova Preov	4		12	10	13	10	45
18.	Juraj Andris	Gym. Sereď	4		10	13	10	11	44
	Andrej Mikulík	Gym. Grösslingová BA	2		9	13	11	11	44
	Miroslav Kačena	Piaristické gym. Trenčín	4	9	8	15	12		44
	Katarína Holeová	Gym. Haanova BA	4		10	12	12	10	44
22.	Slavomír Hoták	Gym. Púchov	3	1	9	9	12	10	41
	Michal Repovský	Gym. tefánika Trebiov	2	10	10	9	12		41
24.	Juraj Porubský	Gym. Farská Nitra	2		5	13	12	10	40
	Peter Klátik	Gym. Grösslingová BA	2		7	11	11	11	40
26.	Luká Poláček	Gym. . túra Modra	2	9	9	9	12		39
	Peter Černo	Gym. udovíta túra Trenčín	1	8	10	9	12		39
28.	Michal Králik	GLN Tomáikova BA	2		12	13	13		38
	Katarina Macova	Gym. Grösslingová BA	2		8	13	7	10	38
30.	Matú Dekánek	Gym. Jura Hronca BA	2		10	15	12		37
	Martin Salaj	Gym. Čadca	3		10	9	8	10	37
	Duan Domány	Gym. P. Horova Michalovce	1		14	13		10	37
33.	Ivan Trejbal	Gym. robárova Koice	2		11	13	12		36
34.	Marek Hanes		1	7	8	13	7		35
35.	Peter Molnár	Gym. Metodova BA	4		10	13	10		33
36.	Marek Dovjak	Gym. Ke« marok	4		12	7	13	0	32
	Marek Tomacha	Gym. sv. Urule BA	4		10	10	12		32
	Jozef Hopko	Gym. Nedo« erského Prievidza	2		12	9	11		32
	Jana Podstupkova	Gym. Grösslingová BA	2		10		12	10	32
40.	Michal Kadák	Gym. udovíta túra Trenčín	2		10	11	10		31
	Daniel Sloboda	Gym. Párovská Nitra	2		8		12	11	31
	Ladislav Máte	SPE Stará Turá	3		10	9	12		31
	Jozef Legeny	G	1		10	13	8		31
44.	Peter Havlíček	Gym. Metodova BA			10	9		11	30
45.	Pavol Szórád	Gym. Farská Nitra	2		5	13		10	28
	Róbert imon	Gym. Hlohovec	2		9		9	10	28
47.	Martin Fiala	Gym. Grösslingová BA	2		8		8	10	26
	Sylvia Leskova	Gym. Metodova BA	4		5	9	12		26
49.	Jana Adamková	Gym. Grösslingová BA	2	2	3	7	2	11	25
50.	Martin Paulich	Gym. Hlohovec			8		6	10	24

	Meno a priezvisko	kola	Trieda	11	12	13	14	15	Σ
51.	Peter Goga	Gym. 17. novembra Topožčany	2		10	9	3		22
	Jakub Steinhauzer	Gym. Metodova BA	4		10		12		22
53.	Hana Hudáková	Gym. Golianova Nitra	1		10			11	21
	Milo Čiernik	Gym. Lettricha Martin	1		9		12		21
	Magdaléna Jurkovičová	Gym. Golianova Nitra			9	12			21
56.	Zsolt Szabó	Gym. Mláde« nícka ahy	4		8	11			19
57.	Matú Svrček	Gym. Stará ubovňa	2		8			10	18
58.	tefan Kritofík	SP Levice	3		12		3		15
59.	Peter Kopčanský	Gym. Krompachy	4					10	10
	Andrej Segeč	Gym. Golianova Nitra	4		10				10
61.	Slavomír Sihelník	Gym. Alejová Koice	4		9				9
62.	Ondrej Kolibiar				4				4