

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 26 (1998/1999)

Številka 6

Stran 355

Martin Juvan:

IŠČEMO BESEDO

Ključne besede: računalništvo, matematika, kombinatorika, faktorielni zapis števil, nizi znakov, sestavljanje besed.

Elektronska verzija: <http://www.presek.si/26/1384-Juvan.pdf>

© 1999 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

IŠČEMO BESEDO

Če imamo na razpolago n različnih črk, lahko iz njih sestavimo $n!$ različnih besed, v katerih vsaka črka nastopa natanko enkrat. Pri tem pojem beseda označuje poljubno (končno) zaporedje znakov. Seveda pa tako zaporedje le redkokdaj določa besedo slovenskega jezika. Iz črk **a**, **h** in **m** lahko tako sestavimo naslednje besede:

ahm, amh, ham, hma, mah, mha.

Vaša naloga je, da v enem od svojih priljubljenih programskih jezikov sestavite program, ki kot vhod vzame skupino n različnih črk in naravno število k ter med $n!$ različnimi besedami, ki jih lahko dobimo z razmeščanjem danih črk, poišče tisto, ki je pri ureditvi po abecedi na k -tem mestu. Pri črkah **a**, **h**, **m** in številu $k = 3$ je to beseda **ham**. Poskusite še, kakšen odgovor dobite pri črkah **a**, **b**, **o**, **r**, **v** in številu $k = 38$.

Martin Juvan

IŠČEMO BESEDO – Rešitev s str. 355

Ljudje števila običajno zapisujemo v desetiškem sestavu. V računalništvu večkrat srečamo tudi osnove dva, osem in šestnajst. Seveda pa obstajajo tudi bolj nenavadni zapisi. Videli bomo, da je rešitev naloge povezana z enim od takih zapisov.

Števila lahko zapišemo tudi v "faktorielnem" zapisu. Vsako naravno število k od 0 do $n! - 1$ lahko enolično zapišemo kot

$$k = c_{n-1} \cdot (n-1)! + c_{n-2} \cdot (n-2)! + \dots + c_2 \cdot 2! + c_1 \cdot 1!,$$

kjer za številke c_i , $i = 1, \dots, n-1$, velja $c_i \in \{0, 1, \dots, i-1, i\}$. O obstoju in enoličnosti zapisa se lahko prepričamo z matematično indukcijo.

In kako poiščemo faktorielni zapis števila? Podobno, kot poiščemo zapise v bolj običajnih sestavih. Dvojiški zapis števila poiščemo npr. tako, da število zaporedoma celoštevilsko delimo z 2 in beležimo ostanke. Deljenje končamo, ko število postane enako 0. Ostanke, prebrani od zadnjega proti prvemu, dajo dvojiški zapis izbranega števila. Npr.:

$$\begin{aligned} 37 &= 18 \cdot 2 + 1 \\ 18 &= 9 \cdot 2 + 0 \\ 9 &= 4 \cdot 2 + 1 \\ 4 &= 2 \cdot 2 + 0 \\ 2 &= 1 \cdot 2 + 0 \\ 1 &= 0 \cdot 2 + 1 \end{aligned}$$

Torej je $37 = 100101_{(2)}$. Posamezna mesta v dvojiškem zapisu so utežena s potenami števila 2: 1, 2, 4, 8, ... V faktorielnem zapisu pa so posamezna mesta utežena s fakultetami števil: 1!, 2!, 3!, 4!, ... Pri iskanju zapisa zato na prvem koraku število delimo z 2, na drugem s 3, na tretjem s 4 itn. Tako dobimo

$$\begin{array}{rcl} 37 &= & 18 \cdot 2 + 1 & k &= & k_2 \cdot 2 + c_1 \\ 18 &= & 6 \cdot 3 + 0 & k_2 &= & k_3 \cdot 3 + c_2 \\ 6 &= & 1 \cdot 4 + 2 & & & \vdots \\ 1 &= & 0 \cdot 5 + 1 & k_{n-1} &= & k_n \cdot n + c_{n-1} \end{array}$$

Če je število k manjše od $n!$, potem je zadnji kvocient k_n enak 0. Faktorielni zapis števila 37 je torej $37 = 1201_{(1)}$.


```

{ Poiščemo zapis števila  $k - 1$  v 'faktorielnem' zapisu. }
k := k - 1;
for i:=1 to n - 1 do begin           { Zapis ima  $n - 1$  štev. }
  stevka[n - i] := k mod (i + 1); { Najprej dobimo zadnjo števko. }
  k := k div (i + 1);
end;

if k > 0 then begin                 { Število  $k$  je preveliko. }
  writeln('Toliko različnih besed ne obstaja. ');
  PoisciBesedo := '';
  exit;
end;

{ Zgradimo iskano besedo. }
beseda := '';
for i:=1 to n - 1 do begin
  { Pazimo, da števke povečamo za ena. }
  beseda := beseda + crke[stevka[i] + 1];
  delete(crke, stevka[i] + 1, 1);
end;
beseda := beseda + crke;           { Dodamo še preostalo črko. }

PoisciBesedo := beseda;
end; { PoisciBesedo }

```

Črke, ki jih imamo na voljo za gradnjo besed, smo predstavili z nizom znakov (tip `string`). Taka izbira nam olajša brisanje že porabljenih črk. Uporabimo lahko kar v turbo pascal vgrajeni podprogram `delete` (ta ima tri parametre: niz, iz katerega brišemo znake, indeks prvega zbrisane znaka in število zbrisanih znakov). Funkcija `PoisciBesedo` zahteva, da so črke v parametru `crke` urejene po abecedi, sicer ne deluje pravilno. Pokličemo jo lahko kar s konstantnim nizom in številom, saj za oba parametra uporabljamo prenos po vrednosti. Tako na primer klic `PoisciBesedo('aborv', 38)` vrne niz `bravo`.

Martin Juvan
