PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652
Letnik **31** (2003/2004)
Številka 1
Strani 22–27

Martin Juvan:

O RAZVRSTITVAH IN PERMUTACIJAH

Ključne besede: kombinatorika, računalništvo, razvrstitve, permutacije.

Elektronska verzija: http://www.presek.si/31/1538-Juvan.pdf

O RAZVRSTITVAH IN PERMUTACIJAH

Pri programiranju se včasih zgodi, da je treba dane objekte razvrstiti na vse možne načine in za vsako razvrstitev preveriti, ali ima neko želeno lastnost. Seveda je tako preverjanje smiselno le za majhne skupine objektov. Če je namreč v skupini n objektov, potem je vseh razvrstitev ravno $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$. Na prvo mesto lahko postavimo katerikoli objekt, na drugo kateregakoli od preostalih $n-1$ itd. Števila $n!$ pa se zelo hitro večajo: $5! = 120$, $10! = 3628800$, $20! \doteq 2.43 \cdot 10^{18}$.

Matematično razvrstitve predstavimo s permutacijami. Naj bo X (končna) množica z n elementi. Da poenostavimo pisavo, običajno vzamemo $X = \{1, 2, \ldots, n\}$. *Permutacije* množice X so natanko bijektivne preslikave iz množice X nazaj v množico X. Množico vseh permutacij označimo z S_X, $S(X)$ ali pa kar z S_n. Naj bo π permutacija množice X, torej bijektivna preslikava $\pi: X \to X$. Če n ni prevelik, jo pogosto podamo s tabelo

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ \pi(1) & \pi(2) & \ldots & \pi(n) \end{pmatrix}.$$

Za razvrstitev, ki jo določa permutacija, vzamemo kar spodnjo vrstico gornje tabele. Vrednost $\pi(1)$ tako pove, kateri objekt je na mestu 1, vrednost $\pi(2)$, kateri objekt je na mestu 2, itd. Možna je tudi nasprotna interpretacija, pri kateri vrednost $\pi(i)$ pove, na katerem mestu v razvrstitvi, ki pripada permutaciji π, se nahaja objekt i.

S permutacijami lahko tudi računamo. Če vzamemo dve permutaciji iste množice, lahko naredimo njun kompozitum (kot kompozitum preslikav) in spet dobimo permutacijo iste množice. Tu smo upoštevali, da je kompozitum bijektivnih preslikav spet bijektivna preslikava. Spomnimo se, da kompozitum preslikav označujemo s $\circ$ in da je definiran kot $(\pi_1 \circ \pi_2)(x) := \pi_1(\pi_2(x))$. Na primer, pri $n = 5$ vzemimo

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 2 & 5 & 1 \end{pmatrix} \quad \text{in} \quad \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 5 & 3 & 4 \end{pmatrix}.$$

Potem je

$$\pi_1 \circ \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{pmatrix} \quad \text{in} \quad \pi_2 \circ \pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 4 & 2 \end{pmatrix}.$$

Mimogrede opazimo, da $\pi_1 \circ \pi_2 \neq \pi_2 \circ \pi_1$. Učeno pravimo, da kompozitum permutacij ni *komutativna* operacija. Je pa kompozitum *asociativna* operacija, kar pomeni, da lahko "prestavljamo oklepaje". Za vsako trojko permutacij $\pi_1, \pi_2, \pi_3 \in S_n$ namreč velja $(\pi_1 \circ \pi_2) \circ \pi_3 = \pi_1 \circ (\pi_2 \circ \pi_3)$. Kot vse bijektivne preslikave imajo tudi permutacije inverze. Če permutacija preslika i v j, potem njen inverz preslika j v i. Inverz permutacije π bomo označili s π^{-1}.

Pri računanju je zelo pomemben zapis permutacij v obliki produkta *ločenih ciklov*. Poglejmo primer. Vzemimo permutacijo $\pi \in S_9$:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 2 & 4 & 1 & 5 & 9 & 3 & 6 & 8 \end{pmatrix}.$$

Opazimo, da π preslika 1 v 7, 7 v 3, 3 v 4 in 4 nazaj v 1. Podobno gre 6 v 9, 9 v 8 in 8 nazaj v 6. Elementa 2 in 5 ostaneta na svojih mestih. Pravimo tudi, da sta *fiksni točki* permutacije. Permutacijo π lahko zapišemo tudi kot

$$\pi = (1\ 7\ 3\ 4) \circ (2) \circ (5) \circ (6\ 9\ 8) = (1\ 7\ 3\ 4)(6\ 9\ 8) = (8\ 6\ 9)(7\ 3\ 4\ 1)\,.$$

Tu zapis $(i_1\, i_2 \ldots i_k)$ označuje *cikel*, permutacijo, ki i_1 preslika v i_2, i_2 v i_3 itn. do i_k, ki gre nazaj v i_1. Zapis permutacije v obliki ločenih ciklov ni enoličen: zapis posameznega cikla lahko začnemo s poljubnim njegovim elementom (npr. $(1\,7\,3\,4) = (7\,3\,4\,1)$), pa tudi cikle lahko navedemo v poljubnem vrstnem redu. Cikli so ločeni, noben element ne nastopa v več ciklih, ampak vsak v natanko enem. V navadi je, da pri zapisu fiksne točke (cikle dolžine 1) izpuščamo. Seveda pa moramo potem za celotni opis permutacije poznati še število n. Prav tako izpuščamo znak za kompozitum med cikli. Kadar imamo permutacijo zapisano kot produkt ločenih ciklov, lahko njen inverz dobimo tako, da vse cikle obrnemo:

$$\pi^{-1} = (1\ 4\ 3\ 7)(6\ 8\ 9)\,.$$

Povejmo še, kako razvrstitev *premešamo* v skladu s permutacijo τ. To storimo tako, da za vsak i od 1 do n objekt, ki je na mestu i, prestavimo na mesto $\tau(i)$. Poglejmo primer. Vzemimo razvrstitev B, D, A, E, C in jo premešajmo v skladu s 3-ciklom $(1\,4\,3)$. Dobimo razvrstitev A, D, E, B, C. Recimo, da začetno razvrstitev določa permutacija π. Katera permutacija določa premešano razvrstitev? Krajši razmislek pokaže, da je to permutacija $\pi \circ \tau^{-1}$. Res, kar poglejmo, kateri objekt je v premešani permutaciji na mestu i. Tisti, ki je na začetku na mestu, ki ga τ preslika v i. To mesto je $\tau^{-1}(i)$, tam pa je objekt $\pi(\tau^{-1}(i)) = (\pi \circ \tau^{-1})(i)$.

Po nekoliko daljšem uvodu smo tako prišli do jedra prispevka. Predstavil bi vam rad štiri desetletja star postopek, s katerim lahko zgradimo vse razvrstitve (oz. permutacije), vsako natanko enkrat. V postopku bomo razvrstitev predstavili z enorazsežno tabelo **p**, pri čemer bodo indeksi v mejah od 1 do n. Kadar bomo nanjo pogledali kot na permutacijo, bo to spodnja vrstica iz dvovrstičnega zapisa permutacije v obliki tabele z začetka prispevka. Splošna shema, v katero sodi naš postopek, je rekurzivna:

```
permutacije(p, n)
  za i = 1, 2, ..., n − 1 ponovi
    če je n > 2, pokliči permutacije(p, n − 1), sicer preglej p
    v p zamenjaj elementa na mestih n in f(n, i)
  do tod
  če je n > 2, pokliči permutacije(p, n − 1), sicer preglej p
```

Pri pravilnem delovanju postopka pričakujemo, da bo na mestih, kjer pregledamo tabelo **p**, v njej vsakič druga razvrstitev. Razvrstitev, ki se v tabeli nahaja ob začetnem klicu, je lahko poljubna, res pa je, da bo postopek to razvrstitev obravnaval kot tisto, ki ustreza identični permutaciji.

Bistvo postopka se skriva v izbiri predpisa f, ki določa, kako med seboj zamenjujemo elemente tabele. Predpis mora zagotoviti, da bomo res zgradili vse razvrstitve. Na prvi pogled pa niti ni jasno, ali tak predpis sploh obstaja. No, B. R. Heap je leta 1963 opazil, da za f lahko vzamemo

$$f(n,i) := \begin{cases} 1, & \text{če je } n \text{ lih,} \\ i, & \text{če je } n \text{ sod.} \end{cases}$$

V nadaljevanju prispevka se bomo poskušali prepričati, da z zgornjo izbiro res dobimo vse razvrstitve. Razmislek bo zahteval kar nekaj računanja s permutacijami.

Poglejmo najprej majhne primere. Če je $n = 1$, se telo zanke sploh ne izvrši, edino razvrstitev pa pregledamo v zadnji vrstici. Pri $n = 2$ se telo zanke izvede enkrat. V njem pregledamo začetno razvrstitev, nato pa zamenjamo elementa na mestih 1 in 2 ($f(2,1) = 1$ in $n = 2$). Po koncu zanke pregledamo še drugo razvrstitev.

V splošnem pa razmišljajmo takole. Če je $n > 2$, klic podprograma permutacije izvede n rekurzivnih klicev, $n-1$ v zanki in enega po koncu zanke. Prepričati se moramo, da se pri rekurzivnih klicih na mestu n zvrstijo prav vsi elementi. Nato upoštevamo predpostavko, da rekurzivni klici (pri njih ima drugi parameter manjšo vrednost) res naredijo vse razvrstitve elementov na mestih od 1 do $n-1$, in tako dobimo vse razvrstitve vseh n elementov. Takemu načinu razmišljanja pravimo dokazovanje z *matematično indukcijo*. *Bazo indukcije*, preverjanje trditve za $n=1$ in $n=2$, smo že premislili v prejšnjem odstavku. Ravnokar pa smo opisali tudi načrt za dokaz *indukcijskega koraka*.

Če želimo razumeti, kako se izmenjujejo elementi na zadnjem mestu, moramo vedeti, kako rekurzivni klic premeša elemente na prvih $n-1$ mestih. Označimo s τ_n permutacijo, v skladu s katero je po klicu permutacije(**p**, n) premešana tabela **p**. S sledenjem postopku (ali pa s pomočjo programa, ki izvaja opisani postopek) ugotovimo, da velja $\tau_2 = (1\,2)$, $\tau_3 = (1\,3)$, $\tau_4 = (1\,4\,3\,2)$, $\tau_5 = (1\,5)$, $\tau_6 = (1\,6\,5\,2\,3\,4)$, $\tau_7 = (1\,7)$ in $\tau_8 = (1\,8\,7\,2\,3\,4\,5\,6)$. Z nekaj smelosti postavimo domnevo, da je permutacija τ_n enaka 2-ciklu $(1\,n)$, če je n lih, in enaka n-ciklu $(1\;n\;n{-}1\;2\;3\;\ldots\;n{-}2)$, če je n sod.

Obnašanje postopka bomo opisali s permutacijami. Naj bo π_i, $1 \le i \le n$, permutacija, ki določa razvrstitev tik pred i-tim rekurzivnim klicem. Iz opisa postopka sledi, da za $i = 1, 2, \ldots, n-1$ velja

$$\pi_{i+1} = \begin{cases} \pi_i \circ \tau_{n-1}^{-1} \circ (1\;n), & \text{če je } n \text{ lih,} \\ \pi_i \circ \tau_{n-1}^{-1} \circ (i\;n), & \text{če je } n \text{ sod.} \end{cases}$$

Prvi kompozitum ustreza rekurzivnemu klicu, drugi pa sami zamenjavi v postopku. Pri tem smo upoštevali, da so 2-cikli kar sami svoji inverzi. Razvrstitev, ki je v tabeli **p** po koncu klica, pa ustreza permutaciji $\pi_n \circ \tau_{n-1}^{-1}$.

Dokazovanje pravilnosti postopka smo tako prevedli na povsem računski, algebraični problem. Pokazati moramo, da so elementi $\pi_i(n)$, $i = 1, \ldots, n$, med seboj različni (kar pomeni, da je pri vsakem rekurzivnem klicu na mestu n drugačen element) in da je $\pi_n \circ \tau_{n-1}^{-1} = \pi_1 \circ \tau_n^{-1}$ (kar pomeni, da je končna razvrstitev ravno s permutacijo τ_n premešana začetna razvrstitev π_1). Brez škode lahko vzamemo, da je π_1 identična permutacija. V računu bomo glede na parnost števila n ločili dve možnosti.

Naj bo n liho število. Potem je

$$\begin{aligned} \pi_2 = \tau_{n-1}^{-1} \circ (1\;n) &= (1\;n{-}3\;n{-}4\;\ldots\;2\;n{-}2\;n{-}1)\,(1\;n) \\ &= (1\;n\;n{-}3\;n{-}4\;\ldots\;2\;n{-}2\;n{-}1) \end{aligned}$$

cikel dolžine n. Velja tudi $\pi_i = \pi_2^{i-1}$. Ciklična permutacija vsak element krožno preslika v njegovega naslednika. Podobno velja za potenco cikla, le da moramo narediti toliko korakov naprej po ciklu, kolikor je vrednost eksponenta. Tako ugotovimo, da velja

$$\begin{array}{c|cccccccc} i & 1 & 2 & \dots & i & \dots & n-3 & n-2 & n-1 & n \\ \hline \pi_i(n) & n & n-3 & \dots & n-i-1 & \dots & 2 & n-2 & n-1 & 1 \end{array}.$$

Ker so v spodnji vrstici tabele sama različna števila, je prvi del trditve dokazan. Poglejmo še izraz $\pi_n \circ \tau_{n-1}^{-1}$. Če upoštevamo, da je k-ta potenca cikla dolžine k identična permutacija, sledi še drugi del trditve:

$$\pi_n \circ \tau_{n-1}^{-1} = \pi_2^{n-1} \circ \tau_{n-1}^{-1} \circ (1\ n) \circ (1\ n) = \pi_2^n \circ (1\ n) = (1\ n) = \tau_n^{-1}.$$

Ostane še možnost, ko je n sodo število. Tedaj je $\tau_{n-1}^{-1} = (1\, n-1)$. Tako z nekaj računanja dobimo

$$\begin{aligned}
\pi_2 &= (1\ n-1) \circ (1\ n) = (n-1\ 1\ n),\\
\pi_3 &= \pi_2 \circ (1\ n-1) \circ (2\ n) = (n-1\ n\ 2),\\
\pi_4 &= \pi_3 \circ (1\ n-1) \circ (3\ n) = (n-1\ 1\ n\ 3\ 2),\\
\pi_5 &= \pi_4 \circ (1\ n-1) \circ (4\ n) = (n-1\ n\ 4\ 3\ 2),\\
\pi_6 &= \pi_5 \circ (1\ n-1) \circ (5\ n) = (n-1\ 1\ n\ 5\ 4\ 3\ 2).
\end{aligned}$$

Natančneje, za lihe $i > 1$ velja

$$\pi_i = (n-1\ \ n\ \ i-1\ \ i-2\ \dots\ 3\ \ 2)\,,$$

za sode $i \neq n$ pa

$$\pi_i = (n-1\ \ 1\ \ n\ \ i-1\ \ i-2\ \dots\ 3\ \ 2)\,.$$

Zadnja permutacija π_n je enaka

$$\pi_n = \pi_{n-1} \circ (1\ \ n-1) \circ (n-1\ \ n) = (1\ n)\,(n-1\ \ n-2\ \dots\ 3\ \ 2)\,.$$

Tako dobimo

$$\begin{array}{c|ccccccccc} i & 1 & 2 & 3 & 4 & \dots & i & \dots & n-1 & n \\ \hline \pi_i(n) & n & n-1 & 2 & 3 & \dots & i-1 & \dots & n-2 & 1 \end{array}.$$

V spodnji vrstici tabele so zopet sama različna števila, tako da prvi del trditve drži. Poračunajmo še $\pi_n \circ \tau_{n-1}^{-1}$:

$$\begin{aligned}\pi_n \circ \tau_{n-1}^{-1} &= (1\ n)\,(n{-}1\ \ n{-}2\ \dots\ 3\ 2)\,(1\ \ n{-}1) \\ &= (1\ \ n{-}2\ \ n{-}3\ \dots\ 3\ 2\ \ n{-}1\ \ n) = \tau_n^{-1}\,.\end{aligned}$$

S tem je pravilnost postopka dokazana.

Opisanega postopka vam gotovo ne bo težko pretvoriti v program, ki bo s pomočjo rekurzije zgradil vse razvrstitve. Bolj izkušenim programerjem pa predlagam, da poskusijo napisati nerekurzivno različico postopka. Ta mora seveda graditi razvrstitve v povsem enakem vrstnem redu kot rekurzivna. Po mojih izkušnjah naj bi bila tudi za slabo petino hitrejša (je pa to seveda odvisno od računalnika in prevajalnika, ki ga uporabljate).

V prispevku smo spoznali Heapov postopek za pregled vseh razvrstitev (oz. permutacij), ki ga je preprosto opisati, precej težje pa se je prepričati, da deluje pravilno. Za dokazovanje pravilnosti smo izbrali bolj algebraično, matematično korektno pot, ki pa ni najbolj nazorna. Smo se pa pri tem vsaj poskušali naučiti osnov računanja s permutacijami.

Martin Juvan

NALOGA ZA SPRETNE (IN POTRPEŽLJIVE) RAČUNARJE – Rešitev s str. 3

Račun pokaže, da se ploščina trikotnika in težiščnici izražajo takole:

$$p = 4x^2(x^4 - 80x^2 + 1024)\,,$$
$$2t_a = (x^2+32)y + 3x(x^2-32) \quad \text{in} \quad 2t_b = (x^2+32)y - 3x(x^2-32)\,.$$

Na desni so polinomi spremenljivk x in y s celimi koeficienti.

Pripomba. V nalogi navedeni izrazi za a, b in c niso pozitivni pri vseh parih števil x, y, ki zadoščajo enačbi $y^2 = x(x^2 + 5x - 32)$. Treba je vzeti trikotnik, ki ima za stranice absolutne vrednosti $|a|$, $|b|$, $|c|$. V Heronovem obrazcu za ploščino in v obrazcih za težiščnice pa nastopajo samo kvadrati stranic in zato morebitni negativni predznaki ne motijo pri računanju. Tudi ploščina p in dvojni težiščnici $2t_a$ in $2t_b$ so, strogo vzeto, absolutne vrednosti izrazov na desni.

Ivan Vidav