Generiranje vseh podmnožic dane množice – leksikografska ureditev

Andrej Taranenko

V prispevku bomo spoznali algoritme, ki se uporabljajo za generiranje vseh možnih podmnožic poljubne dane končne množice v nekem vrstnem redu.

Naj bo n poljubno naravno število in $S = \{1, 2, \ldots, n\}$. Pogledali bomo, kako tvorimo vse podmnožice množice S. Vemo, da je število vseh podmnožic poljubne množice z n elementi enako 2^n. S $\boldsymbol{S}$ označimo seznam[1] vseh podmnožic množice S. Natančneje, $\boldsymbol{S} = [S_0, S_1, \ldots, S_{2^n-1}]$, pri čemer za vsak $i \in \{0, 1, \ldots, 2^n - 1\}$ velja $S_i \subseteq S$ in so vse množice v seznamu $\boldsymbol{S}$ paroma različne. Ker konkreten seznam določa vrstni red elementov v njem, mu rečemo tudi *ureditev* elementov.

Naj bo $\boldsymbol{S} = [S_0, S_1, \ldots, S_{2^n-1}]$ ureditev vseh podmnožic množice S. Ker je s tem določen vrstni red vseh teh podmnožic, ima vsaka podmnožica S_i, za poljubni $i \in \{0, 1, \ldots, 2^n - 2\}$, svojega *naslednika* S_{i+1} - podmnožico, ki se v izbranem vrstnem redu pojavi neposredno za njo. Množica S_{2^n-1} pa nima naslednika oziroma bomo rekli, da je ta *nedefiniran*. Velja torej:

- $\text{naslednik}(S_i) = \begin{cases} S_{i+1}, & 0 \le i < 2^n - 1 \\ \textit{nedefiniran}, & i = 2^n - 1. \end{cases}$

Naj bo $i \in \{0, 1, \ldots, 2^n - 1\}$ in A množica iz seznama $\boldsymbol{S}$. Število i je *rang množice* A natanko tedaj, ko je $A = S_i$. Rang množice A v izbrani ureditvi označimo z $\text{rang}(A)$ in rečemo, da rangiramo množico A. Funkcija $\text{rang} : \boldsymbol{S} \to \{0, 1, \ldots, 2^n - 1\}$ je bijektivna. Njen inverz označimo z derang. Velja torej:

- $\text{rang}(A) = i \Leftrightarrow \text{derang}(i) = A,$

za vse A iz seznama $\boldsymbol{S}$ in vse $i \in \{0, 1, \ldots, 2^n - 1\}$. Računanje $\text{derang}(i)$ imenujemo derangiranje števila i.

Poleg iskanja naslednjega elementa v ureditvi nas bosta zanimala tudi algoritma rangiranja in derangiranja. Še več, naslednika lahko preprosto definiramo s pomočjo rangiranja in derangiranja na naslednji način:

- $$\text{naslednik}(A) = \begin{cases} \text{derang}(\text{rang}(A) + 1), \\ \qquad \text{če } \text{rang}(A) < |\boldsymbol{S}| - 1, \text{ in} \\ \textit{nedefiniran}, \\ \qquad \text{če } \text{rang}(A) = |\boldsymbol{S}| - 1. \end{cases} \tag{1}$$

Primer 1. Poglejmo dve izmed vseh možnih ureditev podmnožic množice $\{1, 2, 3\}$. Prva ureditev naj bo

- $\mathcal{A} = \left[\overset{0}{\{\}}, \overset{1}{\{1\}}, \overset{2}{\{2\}}, \overset{3}{\{3\}}, \overset{4}{\{1,2\}}, \overset{5}{\{1,3\}}, \overset{6}{\{2,3\}}, \overset{7}{\{1,2,3\}}\right],$

druga pa

- $\mathcal{B} = \left[\overset{0}{\{\}}, \overset{1}{\{3\}}, \overset{2}{\{2\}}, \overset{3}{\{2,3\}}, \overset{4}{\{1\}}, \overset{5}{\{1,3\}}, \overset{6}{\{1,2\}}, \overset{7}{\{1,2,3\}}\right].$

Nad posameznimi množicami so zapisani rangi v dani ureditvi. Vidimo lahko, da ima množica $\{\}$ v obeh ureditvah rang enak 0. Množica $\{1\}$ ima v ureditvi $\mathcal{A}$ rang enak 1, v ureditvi $\mathcal{B}$ pa rang enak 4. Za ureditev $\mathcal{A}$ je $\text{derang}(6)$ množica $\{2, 3\}$, za ureditev $\mathcal{B}$ pa je $\text{derang}(6)$ množica $\{1, 2\}$. Naslednik množice $\{1, 2\}$ v ureditvi $\mathcal{A}$ je množica $\{1, 3\}$, v ureditvi $\mathcal{B}$ pa množica $\{1, 2, 3\}$.

[1] V strukturi *seznam* je pomemben vrstni red elementov. Sezname bomo zapisovali med oglate oklepaje.

Podmnožico T množice $S = \{1, 2, \ldots, n\}$ si lahko v računalniku učinkovito predstavimo z binarnim nizom dolžine n, označili ga bomo z $B(T)$. $B(T)$ torej vsebuje n bitov: $b_{n-1}b_{n-2}\ldots b_0$, pri tem za vsak $i \in \{0, 1, \ldots, n-1\}$ velja:

$$b_i = \begin{cases} 0, & \text{če } n - i \notin T \\ 1, & \text{če } n - i \in T. \end{cases}$$

To pomeni, da je i-ti bit z leve (indeks $n - i$) enak 1, če je število i v podmnožici, in enak 0, če število i ni v podmnožici.

Primer 2. Naj bo $S = \{1, 2, 3\}$ in $T = \{1, 2\}$. Če gledamo na T kot na podmnožico množice S, si jo lahko predstavimo s tremi biti ($n = 3$), in sicer $B(T) = 110$. Natančneje, $b_2 = 1$, ker velja $3 - 2 = 1 \in T$, $b_1 = 1$, ker velja $3 - 1 = 2 \in T$, in $b_0 = 0$, ker velja $3 - 0 = 3 \notin T$.

Vsaki podmnožici T množice $S = \{1, 2, \ldots, n\}$ tako priredimo enolično določen niz $B(T)$. Velja tudi obratno, vsak binarni niz dolžine n enolično določa pripadajočo podmnožico množice S. Ni težko preveriti, da na ta način dobimo vse možne binarne nize dolžine n. Še več, na nize $B(T)$ lahko gledamo kot na predstavitve celih števil med 0 in $2^n - 1$ v dvojiškem številskem sestavu. Naraščajoča ureditev teh celih števil, predstavljenih v dvojiškem številskem sestavu, porodi ureditev podmnožic množice S, ki jo imenujemo *leksikografska ureditev*. Rang neke množice $T \subseteq S$ je torej enak vrednosti, ki jo predstavlja $B(T) = b_{n-1}b_{n-2}\ldots b_0$ v dvojiškem številskem sestavu:

$$\text{rang}(T) = \sum_{i=0}^{n-1} b_i 2^i.$$

Derang števila r, kjer je $0 \leq r \leq 2^n - 1$, pa je podmnožica $U \subseteq S$, za katero je $B(U)$ enak predstavitvi števila r v dvojiškem številskem sestavu.

Primer 3. Poglejmo leksikografsko ureditev podmnožic množice $\{1, 2, 3\}$. Torej števila $0 \leq r \leq 2^3 - 1 = 7$ uredimo v naraščajočem vrstnem redu. Za vsako število pogledamo njegovo predstavitev v dvojiškem številskem sestavu, iz česar dobimo pripadajočo podmnožico.

r	dvojiška predstavitev r	pripadajoča množica
0	000	$\{\}$
1	001	$\{3\}$
2	010	$\{2\}$
3	011	$\{2, 3\}$
4	100	$\{1\}$
5	101	$\{1, 3\}$
6	110	$\{1, 2\}$
7	111	$\{1, 2, 3\}$

TABELA 1.

Leksikografska ureditev podmnožic množice $\{1, 2, 3\}$ je torej

$$S = \left[\overset{0}{\{\}}, \overset{1}{\{3\}}, \overset{2}{\{2\}}, \overset{3}{\{2, 3\}}, \overset{4}{\{1\}}, \overset{5}{\{1, 3\}}, \overset{6}{\{1, 2\}}, \overset{7}{\{1, 2, 3\}}\right].$$

Zapišimo še algoritma rangiranja in derangiranja za leksikografsko ureditev podmnožic množice $S = \{1, \ldots, n\}$. Algoritem 1 prikazuje izračun ranga množice $T \subseteq S$ v leksikografski ureditvi. Z zanko **for** gremo skozi vse elemente množice S. V pogojnem stavku preverimo, ali element i pripada dani podmnožici T. S tem v bistvu preverjamo, ali je v $B(T)$ bit b_{n-i} enak 1. Če je, rangu prištejemo ustrezno potenco števila 2. Računamo torej desetiško vrednost števila $B(T)$, ki jo predstavlja dvojiški zapis.

Algoritem 1: LexRangPodmnozice(n,T)

```
r = 0
for i = n, n − 1, ..., 1 do
    if i ∈ T then
        r = r + 2^(n−i)
return r
```

Algoritem 2 prikazuje, kako v leksikografski ureditvi derangiramo število r, pri čemer je $0 \leq r \leq 2^n - 1$. Tudi tokrat zanka **for** predstavlja sprehod čez vseh n elementov množice S. V pogojnem stavku preverimo, ali je najbolj desna števka v dvojiškem zapisu števila r enaka 1. Če je, v množico dodamo →

element, ki ga ta števka (bit) predstavlja. Deljenje števila r z dva in rezanjem na celi del v dvojiški predstavitvi odreže najbolj desno števko, kar omogoča, da v vsaki iteraciji zanke preverjamo vrednost najbolj desne števke.

Algoritem 2: LexDerangPodmnozice(n,r)

$T = \emptyset$
for $i = n, n-1, \dots, 1$ **do**
  if $r \mod 2 = 1$ **then**
    $T = T \cup \{i\}$
  $r = \left\lfloor \frac{r}{2} \right\rfloor$
return T

Algoritma za izračun naslednika ne bomo posebej zapisovali, saj lahko neposredno uporabimo zvezo med naslednikom ter rangiranjem in derangiranjem, predstavljeno s formulo (1).

Primer 4. Ta primer naredite sami. Naj bo $n = 8$ in $T = \{1, 3, 4, 6\}$. S pomočjo algoritma 1 izračunajte $\operatorname{rang}(T)$. Katero množico dobite, če s pomočjo algoritma 2 izračunte $\operatorname{derang}(181)$?

V tem prispevku smo za našo osnovno množico vzeli množico $S = \{1, \dots, n\}$. Kaj pa, če želimo leksikografsko ureditev poljubne množice A z n elementi? V tem primeru zadostuje, da poiščemo bijekcijo $f : A \to S$. Tako lahko poljubno podmnožico $X \subseteq A$ rangiramo kot

- $\operatorname{rang}(X) = \operatorname{rang}(f(X))$.

Podobno derangiramo število r, $0 \le r \le 2^n - 1$, po naslednji formuli:

- $\operatorname{derang}(r) = f^{-1}(\operatorname{derang}(r))$.

Pri tem je f^{-1} inverzna funkcija funkcije f, torej $f(X) = Y$ natanko tedaj, ko je $f^{-1}(Y) = X$, kjer je $X \subseteq A$ in $Y \subseteq S$.

Na začetku smo videli, da obstaja več različnih ureditev podmnožic dane množice, leksikografska je le ena od njih. Na primeru 4 lahko opazimo tudi naslednje. Množica $\{2, 3\}$ ima rang enak 3, množica $\{1\}$ pa rang enak 4. V leksikografski ureditvi podmnožic množice $\{1, 2, 3\}$ imamo torej dve zaporedni množici, ki sta komplementarni (sta med seboj različni, kolikor se le da). Ali obstaja taka ureditev podmnožic množice $\{1, 2, 3\}$, da se bosta dve poljubni zaporedni množici razlikovali za natanko en element? Odgovor je DA! Taka razvrstitev obstaja za poljubno množico $\{1, 2, \dots n\}$, kjer je $n \in \mathbb{N}$. Ampak to je morda zgodba za kdaj drugič. Vas pa izzivam, da najdete tako razvrstitev podmnožic množice $\{1, 2, 3\}$.

Literatura

[1] Donald L. Kreher, Douglas R. Stinson, *Combinatorial algorithms: generation, enumeration, and search*, CRC Press, 1999.

× × ×

Nalogi

Marko Razpet

1. Urejena m-terica $(x_1, x_2, \dots, x_{m-1}, x_m)$, v kateri so koordinate $x_1, x_2, \dots, x_{m-1}, x_m$ naravna števila, je pitagorejska m-terica, če velja

- $x_1^2 + x_2^2 + \dots + x_{m-1}^2 = x_m^2$

Pri tem je $m \ge 3$. Pitagorejska trojica je na primer (3, 4, 5), ker je $3^2 + 4^2 = 5^2$, pitagorejska četverica pa (2, 3, 6, 7), saj je $2^2 + 3^2 + 6^2 = 7^2$. Dokaži, da je za vsako naravno število n peterica

- $(2n, 2n+1, 2n+2, 6n^2+6n+2, 6n^2+6n+3)$

pitagorejska. Poišči tovrstne pitagorejske peterice, katerih koordinate ne presegajo 100.

2. Izračunaj

- $6^2 - 5^2$, $56^2 - 45^2$, $556^2 - 445^2$, $5556^2 - 4445^2$

Nato rezultate posploši na razliko kvadratov oblike

- $\underbrace{55\dots5}_{n}6^2 - \underbrace{44\dots4}_{n}5^2$.

× × ×