5. kapitola

Kombinatorika II –rekurentné formuly, algoritmy „rozdeľuj a panuj“, metóda inklúzie a exklúzie

5.1 Rekurentné vzťahy

Mnohé enumeračné problémy nie sú riešiteľné použitím štandardných kombinatorických metód, ktoré boli prezentované v predchádzajúcej kapitole. Jeden z týchto „obtiažnych“ enumeračných problémov je výpočet počtu binárnych reťazcov dĺžky n, ktoré neobsahujú sekvenciu dvoch núl (t. j. podreťazec ´00´ sa nesmie vyskytovať v týchto binárnuch reťazcoch). Nech symbol a_n vyjadruje počet takýchto binárnych reťazcov. Pomocou jednoduchých úvah (ktoré budú podrobne špecifikované neskoršie) odvodíme formula

$$a_n = a_{n-1} + a_{n-2}$$

Tento výraz sa nazýva *rekurentná formula*, ktorá spolu s počiatočnými hodnotami $a_2 = 3$, $a_3 = 5$ určuje postupnosť $\{a_n\}$. Z tejto postupnosti sme schopní zostrojiť analytickú formulu pre veličiny a_n ako funkciu argumentu n Ako uvidíme neskoršie, tento prístup založený na rekuretnej formule je aplikovateľný na riešenie širokého spektra enumeračných problémov

Príklad 5.1. Nech počet baktérií sa zdvojnásobí každú hodinu. Ak kolónia obsahuje v počiatočnom čase 5 baktérií, koľko ich bude mať za n hodín. Pre takto špecifikovaný systém platí rekurentná formula $a_n = 2a_{n-1}$, ktorá spolu s počiatočnou podmienkou $a_0 = 5$ jednoznačne určuje veličinu a_n pre každé kladné celočíselné n. Postupným použitím rekurentnej formuly dostaneme, $a_1 = 5 \cdot 2$, $a_2 = 2a_1 = 5(2)^2$,..., $a_n = f(n) = 5(2)^n$. Z tohto riešenie vyplýva, že počet baktérií rastie exponenciálne vzhľadom k „časovej“ diskrétnej premennej n.

Definícia 5.1. ***Rekurentná formula*** pre postupnosť $\{a_n\}$ je rovnica, ktorá vyjadruje n-tý člen postupnosti a_n pomocou predchádzajúcich členov postupnosti, $a_{n-1}, a_{n-2}, ..., a_{n-k}$ (k je celé kladné číslo)

$$a_n = f\left(a_{n-k}, a_{n-k+1}, ..., a_{n-1}\right) \tag{5.1}$$

a počiatočných hodnôt prvých k elementov postupnosti, $a_0, a_1, ..., a_{k-1}$. Postupnosť $\{a_n\}$ sa nazýva ***riešenie*** rekurentnej formuly ak jej jednotlivé členy vyhovujú tejto formule.

Existuje úzky vzťah medzi rekuretnou formulou a ***rekurziou***. Rekurzívny algoritmus zostrojuje pre danú hodnotu „premennej“ n riešenie pomocou riešení rovnakého problému pre menšie hodnoty premennej n. Rekurzívny algoritmus sa ukončí, keď sa dosiahnu hodnoty premennej n, pre ktoré riešenie už poznáme (napr. ako počiatočnú hodnotu). Keď analyzujme zložitosť rekurzívneho algoritmu, dostávame rekuretné formule, ktoré špecifikujú počet operácií (obvykle súčinov) potrebných k získaniu riešenia problému s hodnotou premennej n, pomocou počtu operácií potrebných k získaniu problému s menšími hodnotami premennej n.

Príklad 5.2. Študujme jednoduchý príklad rekurzívneho algoritmu pre výpočet faktoriálu, ktorý využíva rekurentnú formulu $n! = n(n-1)!$, s počiatočnou podmienkou 0!=1

$$n! = \begin{cases} 1 & (n=0) \\ n(n-1)! & (n \geq 1) \end{cases}$$

Táto formula je implmentovaná v Pascale pomocou jednoduchej funkcie s rekurzívným volaním svojho názvu.

```
function Factorial(n:integer):integer;
begin if n=0 then Factorial:=1 else
                  Factorial:=n*Factorial(n-1);
end;
```

Príklad 5.3. Nech postupnosť $\{a_n\}$ je riešenie rekuretnej formuly $a_n = a_{n-1} - a_{n-2}$, pre n=2, 3, ... , a nech počiatočné hodnoty sú a_0=3 a a_1=5. Prvé členy postupnosti $\{a_n\}$ sú

n	a_n
0	3
1	5
2	2
3	-3
4	-5
5	-2
6	3
7	5
8	2

Pokúsime sa odvodiť analytickú formulu $a_n = f(n)$.

n	a_n
0	a_0
1	a_1
2	$a_1 - a_0$
3	$a_2 - a_1 = a_1 - a_0 - a_1 = -a_0$
4	$a_3 - a_2 = -a_0 - (a_1 - a_0) = -a_1$
5	$a_4 - a_3 = -a_1 - (-a_0) = -(a_1 - a_0)$
6	$a_5 - a_4 = a_0 - a_1 - (-a_1) = a_0$
7	$a_6 - a_5 = a_0 - (a_0 - a_1) = a_1$
8	$a_7 - a_6 = a_1 - (a_0) = a_1 - a_0$

Z tejto tabuľky vyplýva, že analytická formula, ktorá určuje jednotlivé členy postupnosti má tvar

$$f(3k) = (-1)^k a_0$$

$$f(3k+1) = (-1)^k a_1$$
$$f(3k+2) = (-1)^k (a_1 - a_0)$$

pre k=0, 1, 2, ...

Zložité úrokovanie

Do banky sme uložili 10000 EUR na ročný úrok 11%. Koľko peňazí budeme mať na vklade po 30 rokoch?

Nech veličina P_n je výška vkladu po n rokoch, táto veličina je špecifikovaná rekurentnou formulou

$$P_n = P_{n-1} + 0.11 P_{n-1} = 1.11 P_{n-1}$$

s počiatočnou podmienkou $P_0 = 10000$. Ak rekuretnú rovnicu riešime postupne (iteračne) dostaneme prvé členy postupnosti riešení $\{P_n\}$

$$P_1 = 1.11 P_0$$
$$P_2 = 1.11 P_1 = (1.11)^2 P_0$$
$$P_3 = 1.11 P_3 = (1.11)^3 P_0$$
....
$$P_n = 1.11 P_{n-1} = (1.11)^n P_0$$

Platnosť poslednej formule pre P_n môže byť verifikovaná pomocou matematickej indukcie. Ak položíme n = 30, potom dostaneme

$$P_{30} = (1.11)^{30} P_0 = (1.11)^{30} \cdot 10000 = 228922.97 \text{ EUR}$$

Fibonacciho[1] čísla

Fibonacci v knihe *Liber Abaci* formuloval túto slávnu úlohu: Na ostrove je umiestnený jeden pár zajacov (opačného pohlavia). Zajace sa môžu množiť až vtedy, keď dosiahnu dospelosť v 2 mesiacoch, potom každý mesiac produkujú nový pár zajacov, pozri tabuľku 5.1.

Tabuľka 5.1. Fibonaciho model zajacov na ostrove

mesiac	*reprodukčné páry*	*nereprodukčné páry*	*celkový počet*
1			1
2			1
3			2
4			3
5			5
6			8

Označme počet párov zajacov v mesiaci n symbolom (**Fibonaciho číslom**) F_n, potom populácia zajacov môže byť modelovaná pomocou rekuretných formúl. Na konci prvého

[1] Leonardo Pisano, nazývaný Fibonacci, (1170? – 1240?), taliansky stredoveký matematik, ktorý sa stal známy svojou knihou *Liber Abaci – Kniha o počítadle* (vydanej v r. 1202), v ktorej zoznámil európsku civilizáciu s arabskou a indickou matematikou.

mesiaca počet párov je $F_1 = 1$, pretože tento pár v priebehu druhého mesiaca nemôže mať potomkov, potom taktiež $F_2 = 1$. Počet párov po n mesiacoch je určený rekurentnou formulou

$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 3) \qquad (5.2)$$

spolu s počiatočnými podmienkami. Podľa tejto rovnice, počet párov v mesiaci n sa rovná súčtu počtu párov v predchádzajúcom mesiaci a počet párov z pred dvoch mesiacov (pretože, každý novorodený pár začne plodiť potomkov až po dvoch mesiacoch). Prvých 9 Fibonaciho čísel je uvedených v tabulke 5.2.

Tabuľka 5.2. Fibonaciho čísla

n	F_n
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34

Binárne reťazce neobsahujúce podreťazec ´00´

V predchádzajúcej kapitole boli študované kombinatorické metódy, ktoré sú založené na niekoľkých jednoduchých princípoch a sú schopné riešiť jednoduchšie enumeračné úlohy. Existujú však mnohé enumeračné problémy, ktoré nie sú riešiteľné kombinatorickými metódami. Pre ilustráciu tejto skutočnosti študujme problém enumerácie binárnych reťazcov danej dĺžky n, v ktorých sa nesmú vyskytovať dve nuly za sebou. Označme množinu takýchto reťazcov R_n, pre $n = 2, 3$, tieto množiny obsahujú binárne reťazce

$$R_2 = \{11, 01, 10\}$$

$$R_3 = \{111, 011, 101, 110, 010\}$$

$$R_4 = \left\{ \underbrace{1111, 0111, 1011, 1101, 0101}_{R_4^a}, \underbrace{1110, 1010, 0110}_{R_4^b} \right\}$$

Množinu R_4 zostrojíme pomocou množín R_2 a R_3

$$R_4 = (R_3 \cdot 1) \cup (R_2 \cdot 10) \qquad (5.3)$$

kde symbol ($R_3 \cdot 1$) sa interpretuje tak, že množina reťazcov z R_1 sú rozšírené o symbol ´1´ z pravej strany. Analogickým spôsobom sa interpretuje aj druhý symbol ($R_2 \cdot 10$) z pravej strany (5.1), ako rozšírenie reťazcov z R_2 o podreťazec ´10´ z pravej strany. Podmnožiny ($R_3 \cdot 1$) a ($R_2 \cdot 10$) sú disjunktné (prečo?), potom pre mohutnosť R_4 platí

$$|R_4| = |R_3 \cdot 1| + |R_2 \cdot 10| \Rightarrow |R_4| = |R_3| + |R_2|$$

Podrobnou analýzou môže byť ukázané, že formula (5.3) môže byť zovšeobecnená do tvaru

$$R_i = (R_{i-1} \cdot 1) \cup (R_{i-2} \cdot 10) \qquad (5.4)$$

Táto formula nám poskytuje predpis, ako tvoriť binárne reťazce dĺžky i, ktoré neobsahujú podreťazec ´00´. Pre mohutnosti množín z (5.2) platí

$$|R_i| = |R_{i-1}| + |R_{i-2}| \qquad (i = 4, 5, 6, \ldots) \qquad (5.5)$$

kde počiatočné hodnoty mohutností sú $|R_2| = 3, |R_3| = 5$. Aby sme zjednodušili formule, zavedieme označenie $a_i = |R_i|$, potom z (5.3) dostaneme ***rekuretnú formulu***, ktorá určuje mohutnosti ďalších množín R_i

$$a_i = a_{i-1} + a_{i-2} \qquad (i = 4,5,6,...) \tag{5.6}$$

pre $a_2 = 3, a_3 = 5$. Postupným použitím tejto formule dostaneme číselné hodnoty veličín a_i, ktoré sú uvedené v tabuľke

i	a_i	i	a_i
2	3	7	34
3	5	8	55
4	8	9	89
5	13	10	144
6	21	...	...

Našim konečným cieľom je zostrojiť analytickú formulu $a_i = f(i)$, ktorá špecifikuje počet binárnych reťazcov dĺžky i, ktoré neobsahujú podreťazec ´00´. Postupným aplikovaním rekurentnej formule (5.4) dostaneme

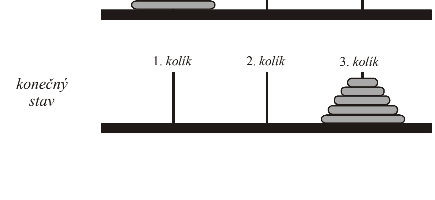
i	a_i	i	a_i
2	3	7	$3a_2 + 5a_3$
3	5	8	$5a_2 + 8a_3$
4	$a_2 + a_3$	9	$8a_2 + 13a_3$
5	$a_2 + 2a_3$	10	$13a_2 + 21a_3$
6	$2a_2 + 3a_3$	...	...

Ak porovnáme v tejto tabuľke čísla pred počiatočnými koeficientmi a_2 a a_3 s Fibonaciho koeficientmi z tabuľky 5.2, dostaneme všeobecná formula pre a_n

$$a_n = F_{n-4}a_2 + F_{n-3}a_3 \quad (pre\ n \geq 4) \tag{5.7}$$

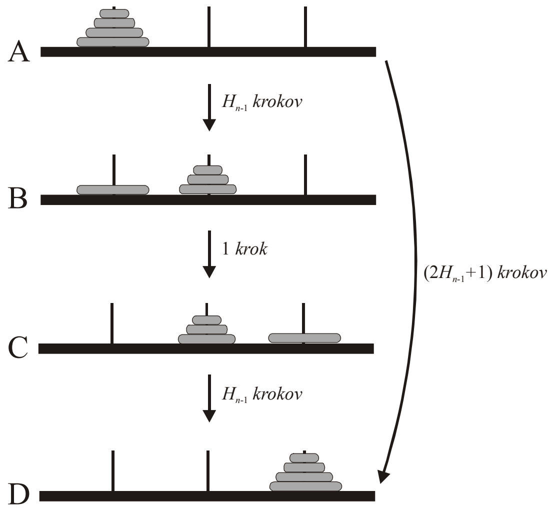
Hanojské veže

Populárna hračka z konca 19. storočia, ktorá bola navrhnutá francúzskym matematikom Èdouardom Lucasom (1842-1891). Princípy tejto hry sú vysvetlené na obr. 5.1. Hra obsahuje n diskov, ktoré sú v počiatočnej pozícii (stave) umiestnené na prvom kolíku. Daný stav sa nazýva prípustný vtedy, ak na ľubovoľnom kolíku nie je umiestnený väčší disk na menšom. Toto ohraničenie nám aj predpisuje aké ťahy (elementárne operácie prenesenia vrchného disku z kolíka na iný kolík) môžu byť použité, aby sme dosiahli konečný stav (cieľovú pozíciu), v ktorej sú všetky disky umiestnené na treťom kolíku. Našou úlohou je zistiť koľko potrebujeme ťahov na to, aby sme v hre obsahujúcej n diskov pretransformovali počiatočný stav na konečný stav[2].



[2] Legenda, ktorá sprevádza hru „hanojské veže", hovorí o tom, že v budhistickom chráme, umiestnenom na tajnom mieste v džungli vo vietnamských horách, sa snažia mnísi ukončiť túto hru obsahujúcu 64 zlatých diskov už mnoho storočí. Veria, že ak sa im podarí hru dokončiť (t. j. dosiahnuť konečný stav), nastane koniec sveta.

Obrázok 5.1. Hra "hanojské veže" obsahuje základnú dosku, na ktorej sú umiestnené tri kolíky. Obsahuje n kruhových diskov s rôznym priemerom a otvorom v strede, pomocou ktorého sa dajú navlieknuť na kolík. *Povolené pozície* kruhových diskov sú také, že na kolíku nie je umiestnený väčší disk na menšom disku. Rozlišujeme dve pozície, východisková pozícia a konečná pozícia. Hra spočíva v úlohe, pomocou postupného premiestňovanie diskov pretransformovať počiatočný stav na konečný cez postupnosť povolených pozícií.



Obrázok 5.2. Enumerácia počtu krokov potrebných k pretransformovaniu počiatočnej pozície (diagram A) na konečnú pozíciu (diagram D). Prvý medzistav (diagram B) je vytvorený tak, že pomocou H_{n-1} krokov presunieme n-1 diskov na druhý kolík, pričom najväčší disk zostáva na prvom kolíku. V ďalšom kroku (diagram C) presunieme najväčší disk na tretí kolík. Tento medzistav je pretransformovaný pomocou H_{n-1} krokov na konečný stav. To znamená, aby sme premiestnili n diskov z prvého kolíka na tretí kolík cez povolené pozície, musíme vykonať aspoň $2H_{n-1}+1$ krokov.

Ukážeme jednoduchý postup pre konštrukciu rekurentnej formuly, ktorá špecifikuje minimálny počet krokov potrebných k tomu, aby sme pretransformovali počiatočnú pozíciu na konečnú pozíciu. Túto veličinu pre hru obsahujúcej n diskov označíme H_n. Postup je znázornený na obr. 5.2, kde sa predpokladá, že hru vieme riešiť pre n-1 diskov. Potom počiatočný stav A pretransformujeme pomocou H_{n-1} ťahov na medzistav B, v ktorom n-1 diskov je premiestnených na druhý kolík, pričom najväčší disk zostáva na pôvodnom prvom kolíku. Nasledujúci medzistav C dosiahneme tak, že najväčší disk premiestnime na tretí kolík. V poslednej časti hry dosiahneme koncový stav D pomocou H_{n-1} ťahov. Poznamenajme, že transformácie A→B a C→D sú z pohľadu hry obsahujúcej n-1 diskov ekvivalentné, postup, ktorý vedie na prvú transformáciu je možné použiť aj k druhej transformácii. Týmto sme ukázali, že potrebný počet krokov na transformáciu A→D pre n diskov je určený jednoduchou rekuretnou formulou

$$H_n = 2H_{n-1} + 1 \tag{5.8}$$

s počiatočnou podmienkou $H_1 = 1$, ktorá špecifikuje počet ťahov potrebných na riešenie hry obsahujúcej jeden disk[3].

Rekurentnú rovnicu (5.8) budeme riešiť pomocou iteračného postupu

---

[3] Pre čitateľa, ktorý má problémy s akceptovaním hry obsahujúcej len jeden disk, môžeme počiatočnú podmienku formulovať pomocou $H_2 = 3$, ktorá špecifikuje počet ťahov pre dva disky. Prvý ťah premiestni menší disk na druhý kolík, druhý ťah premiestni väčší disk na tretí kolík, tretí ťah premiestni menší disk na tretí kolík.

$$
\begin{aligned}
H_n &= 2H_n + 1 \\
&= 2\left(2H_{n-1}+1\right)+1 = 2^2 H_{n-1} + 2 + 1 \\
&= 2^2\left(2H_{n-2}+1\right)+2+1 = 2^3 H_{n-2} + 2^2 + 2 + 1 \\
&\quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&= 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0 \\
&= 2^n - 1
\end{aligned}
\tag{5.9}
$$

kde na záver sme použili formulu pre sumáciu prvých n členov geometrickej postupnosti

$$1 + q + q^2 + \dots + q^n = \frac{q^{n+1}-1}{q-1}$$

ktorá sa jednoducho dokáže tak, že ju vynásobíme členom (q - 1) a vykonáme naznačené súčiny na ľavej strane. Toto bol priamy dôkaz formuly[4] $H_n = 2^n - 1$, keď už poznáme riešenie rekurentnej formuly (5.8), môžeme opakovane dokázať túto formulu pomocou matematickej indukcie.

Príklad 5.4. Enumeračnú metódu pre úlohy hanojských veží môžeme aj použiť pre implementáciu rekurzívneho algoritmu na riešenie tejto úlohy. Algoritmus obsahuje tri základné kroky:

1. krok: prenesenie n-1 diskov z prvého kolíka na druhý kolík,
2. krok: prenesenie jedného kotúča z prvého kolíka na tretí kolík,
3. krok: prenesenie n-1 diskov z druhého kotúča na tretí kotúč.

Z týchto 3 krokov prvý a tretí sú rovnaké úlohy ako pôvodná, iba s menším počtom kolíkov. Druhý krok je elementárny, je to len presun jedného disku. Kroky 1. a 3. riešime tak, že ich znovu rozložíme do troch podkrokov, ale už s menším počtom diskov. Končíme vtedy, keď vyriešime podobným rozkladom do 3 častí ako pôvodnú úlohu, dokiaľ nedostaneme elementárne kroky. Jednoduchá implementácia v Pascale je znázornená v nasledujúcom algoritme:

```
program TowerOfHanoi;
var n : integer;

procedure Hanoi(n,Tfrom,Tto,Tusing : integer);
begin if n > 0 then
      begin Hanoi(n-1,Tfrom,Tusing,Tto);
            writeln('move',Tfrom:1,' -->',Tto:1);
            Hanoi(N-1,Tusing,Tto,Tfrom);
      end;
end;

begin write('n =?');
      readln(n);
      writeln;
      Hanoi(n,1,3,2)
end.
```

[4] Môžeme sa teraz vrátiť k mýtu z druhej poznámky pod čiarou. Minimálny počet ťahov, ktoré potrebujú mnísi na riešenie úlohy s 64 diskami je 2^{64} – 1 = 18446744073709551615 ≈ 1.84 10^{19}. Pre jednoduchosť predpokladajme, že jeden ťah trvá jednu sekundu, potom na riešenie hry potrebujeme aspoň 500 miliard rokov, čo viac ako 30 násobne prekračuje súčasný odhad trvania nášho vesmíru.

Enumerácia hesiel

Predpokladajme pre názornosť, že vstup do počítačového systému je povolený heslom, ktoré má 10 číslic, pričom obsahuje párny počet núl. Tak napríklad reťazec 3214043201 môže byť použitý ako platné heslo, zatiaľ čo reťazec 1020034387 nie je platné heslo. Nech a_n je počet platných hesiel dĺžky n. Počiatočná hodnota nech je $a_1 = 9$, pretože existuje 10 číselných reťazcov, x ktorých jeden, ´0´ nie je platný. Odvodíme rekurentnú formulu pre a_n tak, že budeme študovať možnosť, ako vnikne platný reťazec dĺžky n z reťazcov dĺžky n-1. Existujú dva spôsoby tvorby platných reťazcov:

(1) Platný reťazec dĺžky n-1 rozšírime o jedno nenulové číslo, toto rozšírenie môže byť urobené 9 rôznymi spôsobmi, t. j. príspevok k a_n má tvar $9a_{n-1}$.

(2) Reťazec dĺžky n-1, ktorý nie je platný, je rozšírený o nulu. Tento spôsob tvorby reťazca vytvára z neplatného reťazca platný reťazec, pričom príspevok k a_n sa rovná počtu neplatných reťazcov dĺžky n-1. Pretože celkový počet reťazcov (platných a aj neplatných) dĺžky n-1 sa rovná 10^{n-1}, potom 10^{n-1}-a_{n-1} neplatných reťazcov dĺžky n-1.

Potom, celkový počet platných reťazcov dĺžky n je určený rekurentnou formulou

$$a_n = 9a_{n-1} + \left(10^{n-1} - a_{n-1}\right) = 8a_{n-1} + 10^{n-1} \qquad (5.10)$$

Iteračným riešením tejto rovnice dostaneme

$$\begin{aligned}
a_n &= 8a_{n-1} + 10^{n-1} \\
&= 8\left(8a_{n-2} + 10^{n-2}\right) + 10^{n-1} = 8^2 a_{n-2} + 8\cdot 10^{n-2} + 10^{n-1} \\
&= 8^2\left(8a_{n-3} + 10^{n-3}\right) + 8\cdot 10^{n-2} + 10^{n-1} = 8^3 a_{n-3} + 8^2\cdot 10^{n-3} + 8\cdot 10^{n-2} + 10^{n-1} \\
&= 8^3\left(8a_{n-4} + 10^{n-4}\right) + 8^2\cdot 10^{n-3} + 8\cdot 10^{n-2} + 10^{n-1} = \\
&= 8^4 a_{n-4} + 8^3\cdot 10^{n-4} + 8^2\cdot 10^{n-3} + 8\cdot 10^{n-2} + 10^{n-1}
\end{aligned}$$

Použitím formuly pre sumáciu geometrického radu získame analytickú formulu pre $f(n)$

$$\begin{aligned}
a_n &= 8^{n-1}a_1 + 8^{n-2}\cdot 10^1 + 8^{n-3}\cdot 10^2 + \ldots + 8\cdot 10^{n-2} + 10^{n-1} \\
&= 8^{n-1}a_1 + 8^{n-1}\left(1 + \frac{10}{8} + \frac{10^2}{8^2} + \ldots + \frac{10^{n-1}}{8^{n-1}}\right) - 8^{n-1} \\
&= \frac{8^n + 10^n}{2}
\end{aligned}$$

Tento výsledok môže byť verifikovaný matematickou indukciou.

5.3. Metóda „rozdeľuj a panuj“

Metóda rozdeľuj a panuj patrí medzi efektívne prístupy k riešeniu niektorých typov rekuretných formúl. Predpokladajme, že rekurzívny algoritmus *rozdeľuje* problém dimenzie n na a podproblémov, pričom dimenzia podproblému nech je špecifikovaná zlomkom n/b. Predpokladajme taktiež, že potrebujeme $g(n)$ dodatočných operácií k tomu, aby sme v *unifikačnej* (panovačnej) fáze algoritmu pospájali riešenia podproblémov do celkového riešenia. Potom, ak $f(n)$ vyjadruje počet operácií potrebných na riešenie celkového problému dimenzie n, funkcia f je špecifikovaná formulou nazývanou ***rekuretná formula rozdeľuj–a–panuj***[5]

$$f(n) = af(n/b) + g(n) \tag{5.11}$$

V nasledujúcich ilustračných príkladoch ukážeme aplikácie tejto metódy k riešeniu problémov výpočtovej zložitosti niektorých jednoduchých algoritmov.

Binárne prehľadávanie

Tento typ prehľadávania je založený na tom, že ak hľadáme nejaký element x v neklesajúcej postupnosti čísel $\{a_i\}_{i=1}^{n}$, v prvom kroku testujeme podmienku $x \le a_{n/2}$; ak platí (neplatí), potom hľadáme element x v prvej polovici postupnosti $\{a_i\}_{i=1}^{i=n/2}$ (v druhej polovici postupnosti $\{a_i\}_{i=n/2+1}^{i=n}$). Algoritmus vyžaduje vo všeobecnosti dve porovnania, prvé je porovnanie $x \le a_{n/2}$, druhé porovnanie vykonávame vtedy, ak prvé neplatí, porovnáme $x \le a_n$, ak táto podmienka neplatí, potom automaticky vieme, že x nie je elementom postupnosti $\{a_i\}_{i=1}^{n}$. Potom, ak $f(n)$ je počet porovnaní potrebný na riešenie binárneho prehľadávania dimenzii n

$$f(n) = f(n/2) + 2 \tag{5.12}$$

kde pre jednoduchosť predpokladáme, že n je párne.

Jednoduché riešenie rekurentnej rovnice (5.12), popisujúce časovú zložitosť binárneho prehľadávania, získame za predpokladu, že dimenzia problému je $n = 2^k$, pričom $f(2) = 2$. Iteračným riešením (5.12) dostaneme

$$f(2^k) = 2k \tag{5.13}$$

To znamená, že časová zložitosť je lineárna vzhľadom k exponentu k. Veličina k je určená formulou $k = ln\, n/ln\, 2$, potom približná formula pre zložitosť algoritmu má tvar

$$f(n) = o(ln\, n) \tag{5.14}$$

ktorá je interpretovaná tak, že uvádzame len dominantnú funkčnú závislosť od dimenzii n, ktorá špecifikuje nárast zložitosti pri zväčšovaní n. Z (5.14) vyplýva, že zložitosť binárneho prehľadávania rastie „pomaly“ ako $ln\, n$, t. j. ešte pomalšie ako lineárna závislosť $o(n)$.

Hľadanie minimálneho prvku v postupnosti

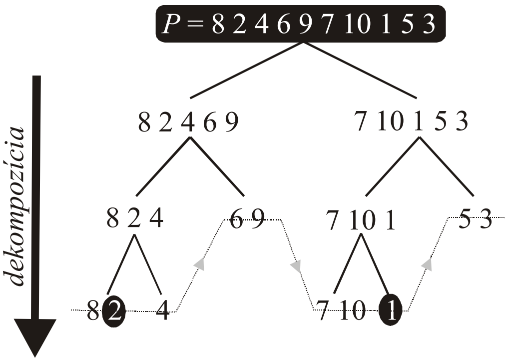
Uvažujme neusporiadanú postupnosť $P = \{a_i\}_{i=1}^{i=n}$, našim cieľom je nájsť minimálny prvok

$$a_{min} = \min_i \{a_i\} \tag{5.15}$$

Zložitosť (počet potrebných operácií – porovnaní) algoritmu pre riešenie tejto úlohy označíme $f(n)$. V prípade, že n = 1, potom $f(1) = 0$, t. j. nepotrebujeme žiadnu operáciu porovnania, element a_1 je automatický minimálnym elementom. Nech $n > 1$, potom postupnosť $\{a_i\}_{i=1}^{i=n}$ rozdelíme na dve polovice. Ak týmto delením pôvodnej postupnosti vznikne postupnosť, ktorá má dĺžku dĺžku 2 alebo 1, potom jednoduchým porovnaním zistíme jej minimálny element a ten porovnáme s hľadaným minimálnym elementom celej postupnosti, ak je menší tak celkové riešenie nahradíme týmto aktuálnym riešením. Tento proces sa rukurzívne

[5] Odborná americká angličtina obsahuje množstvo názvov, ktoré nášmu uchu znejú veľmi exotický a nekonvenčne. Zrejme, nemá zmysel zavádzať pre túto americkú odbornú terminológiu našu špecifickú slovenskú terminológiu, ktorá by bola viac konvenčnejšia a menej exotická, avšak potom by sme museli vytvárať slovníky slovenskej odbornej terminológie s uvedením jej ekvivalentov vo svetových jazykoch.

opakuje a je ukončení vtedy, keď sa dosiahla dĺžka postupnosti 1 alebo 2. Ilustratívny príklad tohto algoritmu je znázornený na obr. 5.3, kde rekurzívne delíme postupnosť dĺžky 10.



Obrázok 5.3. Znázornenie rekurzívneho delenia intervalu dĺžky 10, v prípade, že algoritmus vytvorí postupnosť dĺžky 1 alebo 2, tak túto testujeme či neobsahuje minimálny element (znázornené čiernymi elipsami).

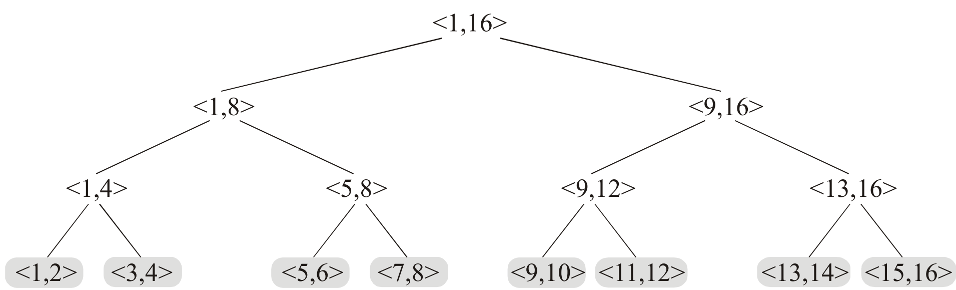
Implentácia rekurzívneho algoritmu pre delenie postupností až na minimálnu dĺžku 1 alebo 2, je uvedená v pseudoPascale:

```
procedure divide(il,ir : integer);
begin Δ:=(ir-il+1) div 2;
      jr:=il+Δ-1; jl:=jr+1;
      Δ1:=jr-il+1; Δ2:=ir-jl+1;
      if Δ1≤2 then
      begin test interval<ail,ajr> for min. element end;
      if Δ2≤2 then
      begin test interval<ajl,air> for min. element end;
      if Δ1>2 then divide(il,jr);
      if Δ2>2 then divide(jl,ir);
end;
```

Táto rekurzívna procedúra postupne delí interval $\langle i_l, i_r \rangle$, kde $\left| i_r - i_l \right| \geq 2$, na dva podintervaly $\langle i_l, j_r \rangle$ a $\langle j_l, i_r \rangle$ tak, že $j_l = j_r + 1$. V prípade, že interval vyhovuje podmienke $\left| i_r - i_l \right| \leq 1$, potom delenie tohto intervalu už neprebieha, ale elementy postupnosti a_{i_r}, a_{i_r} sú testované, či ich minimálna hodnota nie je menšia ako aktuálna hodnota minima

$$\left(min\left\{ a_{i_l}, a_{i_r} \right\} < a_{min} \right) \Rightarrow \left(a_{min} = min\left\{ a_{i_l}, a_{i_r} \right\} \right) \qquad (5.16)$$

Príklad delenia intervalu $\langle 1,10 \rangle$ algoritmom `divide` je znázornený na obr. 5.4.



Obrázok 5.4. Rekurzívny rozklad intervalu $\langle 1,16 \rangle$ na elementárne intervaly dĺžky 2.

Pre jednoduchosť predpokladajme, že $n = 2^k$, potom interval $\langle 1, n \rangle$ môžeme rozdeliť na 2k elementárnych intervalov, pre každý elementárny interval potrtebuje dve porovnania, aby sme vyriešili problém (5.16) (jedno porovnanie pre riešenie problému $min\left\{a_{i_l}, a_{i_r}\right\}$ a druhé porovnanie pre $min\left\{a_{i_l}, a_{i_r}\right\} < a_{min}$). Potom zložitosť algoritmu je určená formulou

$$f(n) = 4k = o(ln\, n) \tag{5.17}$$

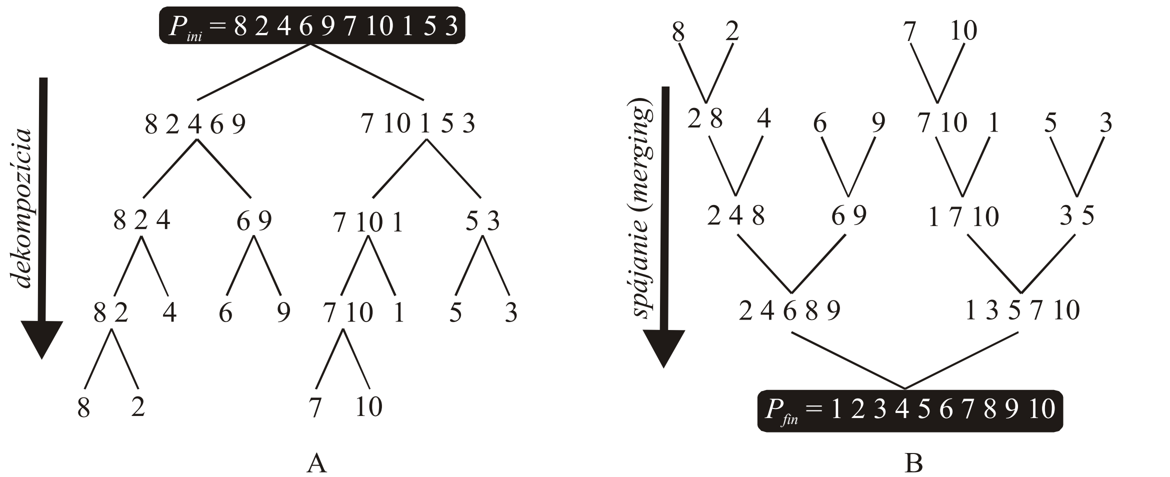
Toto je pomerne prekvapujúci výsledok, že už pre tak jednoduchý algoritmický problém nájdenia minimálneho elementu v postupnosti obsahujúcej n členov, existuje algoritmus, ktorého zložitosť rastie pomalšie ako lineárne vzhľadom k jeho veľkosti. Klasické riešenie tohto problému spočíva v jednoduchom algoritme, v ktorom každý člen postupnosti je porovnaní s minimálnym elementom postupnosti, jeho zložitosť je

$$f(n) = o(n)$$

Použitím metódy „rozdeľuj a panuj", založenej na rekurzívnom delení intervalu, zložitosť algoritmu klesne z lineárnej závislosti na logaritmickú, čo môže mať pre veľké hodnoty n podstatný význam pre akceleráciu nájdenia minimálnoho prvku v postupnosti.

Usporiadanie so spájaním (merge sort)

Cieľom tohto algoritmu je usporiadať postupnosť $P = \{a_i\}_{i=1}^{i=n}$ tak, aby bola neklesajúca, $a_1 \le a_2 \le a_3 ...$. Algoritmus rozdelí postupnosť P na dve polovice P_1 a P_2, každá polovica sa nezávisle usporiada a tieto usporiadané podpostupnosti spojíme (mergujeme) do novej usporiadanej postupnosti. Algoritmus má rekurzívny charakter, v prvej fáze dochádza k rekurzívnemu delenie postupnosti na polovičné časti, ak podpostupnosti obsahujú len jeden element, toto delenie sa zastaví a začne prebiehať opačné rekurzívne spájanie so súčasným usporiadaním, pozri obr. 5.3.



Obrázok 5.3. Diagramatická reprezentácia metódy usporiadania so spájaním. Diagram A znázorňuje prvú fázu algoritmu v ktorej dochádza k postupnému rozkladu neusporiadanej postupnosti na časti, ktoré obsahujú po jednom objekte. Diagram B vyjadruje druhú fázu algoritmu, keď dochádza k postupnému spájaniu fragmentov za súčasného usporiadania spojeného výsledku.

Pre efektívnu implementáciu tohto algoritmu je dôležitá operácia spájania usporiadaných postupností do väčších usporiadaných postupností. Predpokladajme, že máme dve usporiadané postupnosti P_1 a P_2, ktoré spojíme do novej usporiadanej postupnosti P, základné princípy algoritmu sú znázornené v tabule 5.3. V tejto tabuľke čiernymi písmenami sú označené tie číslice, ktoré sú prenášané sprava do postupnosti P (ktorá bola pri zahájení algoritmu prázdna). Na záver algoritmu (riadok 6 v tabuľke), postupnosť P obsahuje spojené postupnosti P_1 a P_2, v usporiadanom - neklesajúcom tvare.

Tabuľka 5.3. Usporiadané spájanie dvoch postupností P_1=(2,3,5,6) a P_1=(1,4)

#	postupnosť P_1	postupnosť P_2	spojená postupnosť P	porovnanie
1	2 3 5 6	1 4		1<2
2	2 3 5 6	4	1	2<4
3	3 5 6	4	1 2	3<4
4	5 6	4	1 2 3	4<5
5	5 6		1 2 3 4	
6			1 2 3 4 5 6	

Z formulácie algoritmu spájanie postupností vyplýva je dôležitá vlastnosť, že spojenie dvoch usporiadaných postupností P_1 a P_2, dĺžky m resp. n, do usporiadanej postupnosti P, ***vyžaduje menej ako m+n porovnaní***. Táto vlastnosť popísaného algoritmu spájania vyplýva zo skutočnosti, že ak obe postupnosti sú usporiadané, potom k nájdeniu minimálneho prvku stačí porovnávať len prvé (najmenšie) elementy oboch postupností P_1 a P_2.

Nech symbol $f(n)$ špecifikuje počet porovnaní, ktorý je potrebný, aby postupnosť $\{a_i\}_{i=1}^{i=n}$ bola usporiadaná do neklesajúcej postupnosti. Podľa algoritmu usporiadania so spájaním, postupnosť $\{a_i\}_{i=1}^{i=n}$ môžeme usporiadať tak, že ju rozdelíme na dve postupnosti $\{a_i\}_{i=1}^{i=n/2}$ a $\{a_i\}_{i=n/2+1}^{i=n}$. Tieto podpostupnosti sa usporiadajú a potom sa spoja do spoločnej usporiadanie postupnosti, pričom tento posledný krok vyžaduje $2(n/2)-1$ porovnaní, potom

$$f(n)=2f(n/2)+n \tag{5.18}$$

Táto rekurentná formula spolu s počiatočnou podmienkou $f(2) = 1$ tvorí základ pre odhad výpočtovej zložitosti algoritmu usporiadania so spájaním. Pre jednoduchosť položme $n = 2^k$, potom iteračným riešením (5.18) dostaneme

$$\begin{aligned} f\left(2^k\right) &= 2f\left(2^{k-1}\right)+2^k \\ &= 2\left(2f\left(2^{k-2}\right)+2^{k-1}\right)=2^2 f\left(2^{k-2}\right)+2^k \\ &= 2^2\left(2f\left(2^{k-3}\right)+2^{k-2}\right)+2^k = 2^3 f\left(2^{k-3}\right)+2\cdot 2^k \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &= 2^k f(1)+(k-1)\cdot 2^k = k\cdot 2^k \end{aligned}$$

Na základe tohto výsledku môžeme zložitosť metódy usporiadania so spájaním odhadnúť takto

$$f(n)=o(n\ln n) \tag{5.19}$$

Jednoduché metódy usporiadania sú obvykle výpočtovo náročne kvadraticky vzhľadom k veľkosti postupnosti, $f(n)=o\left(n^2\right)$, takže výsledok (5.19) možno pokladať za viac ako uspokojivý.

5.4. Princíp inklúzie a exklúzie

V teórii množín (pozri kap. 2.1 a formulu (2.17)) bola diskutované formula pre mohutnosť množiny špecifikovanej ako zjednotenie podmnožín $A_1, A_2, ..., A_n$, uvedieme je spolu s jej duálnou formou

$$\left|A_1 \cup A_2 \cup ... \cup A_n\right| = \sum_{i=1}^{n}\left|A_i\right| - \sum_{\substack{i,j=1\\(i<j)}}^{n}\left|A_i \cap A_j\right| + \sum_{\substack{i,j,k=1\\(i<j<k)}}^{n}\left|A_i \cap A_j \cap A_k\right| + ... \quad (5.20a)$$
$$(-1)^{n+1}\left|A_1 \cap A_2 \cap ... \cap A_n\right|$$

$$\left|\overline{A}_1 \cap \overline{A}_2 \cap ... \cap \overline{A}_n\right| = |U| - \sum_{i=1}^{n}\left|A_i\right| + \sum_{\substack{i,j=1\\(i<j)}}^{n}\left|A_i \cap A_j\right| - \sum_{\substack{i,j,k=1\\(i<j<k)}}^{n}\left|A_i \cap A_j \cap A_k\right| + ... \quad (5.20b)$$
$$(-1)^{n}\left|A_1 \cap A_2 \cap ... \cap A_n\right|$$

kde U je univerzálna množina nad ktorou sú definované množiny A_i. Druhá formula je duálnou formou prvej formuly (5.20a) a je vzťahu,

$$\left|\overline{A}_1 \cap \overline{A}_2 \cap ... \cap \overline{A}_n\right| = \left|\overline{A_1 \cup A_2 \cup ... \cup A_n}\right| = |U| - \left|A_1 \cup A_2 \cup ... \cup A_n\right| \quad (5.21)$$

kde pravá strana bola upravené použitím všeobecného De Morganovho vzťahu pre komplement zjednorenia množín a taktiež použitím jednoduchej formuly pre mohutnosť komplementu množiny $\left|\overline{A}\right| = |U| - |A|$.

Postulujme pre elementy univerza U vlastnosti $p_1, p_2, ..., p_n$, potom vzhľadom k týmto vlastnostiam definujme n množín $A_1, A_2, ..., A_n \subset U$ tak, že elementy množiny A_i majú vlastnosť p_i (pre i = 1, 2, ..., n)

$$A_i = \{x \in U; vlastnosť(x) = p_i\} \quad (5.21)$$

Potom, zjednotenie $A_i \cup A_j$ obsahuje elementy, ktoré majú vlastnosť p_i ***alebo*** vlastnosť p_j. Analogicky, prienik $A_i \cap A_j$ obsahuje elementy, ktoré majú súčasne vlastnosť p_i ***a*** vlastnosť p_j. Táto terminológia je jednoducho zovšeobecniteľná aj pre prieniky alebo zjednotenia viac ako dvoch množín A_i. Komplement

$$\overline{A}_i = \{x \in U; vlastnosť(x) = \overline{p}_i\} \quad (5.22)$$

obsahuje elementy, ktoré *nemajú* vlastnosť p_i, čo je označené symbolom $\overline{p}_i$. Podobne, ako v texte za (5.21) , zjednotenie $\overline{A}_i \cup \overline{A}_j$ obsahuje elementy, ktoré nemajú vlastnosť p_i ***alebo*** nemajú vlastnosť p_j. Prienik $\overline{A}_i \cap \overline{A}_j$ obsahuje elementy, ktoré nemajú súčasne vlastnosť p_i ***a*** vlastnosť p_j.

Nech symbol $N\left(p_{i_1}, p_{i_2}, ..., p_{i_k}\right)$ špecifikuje počet elementov, ktoré majú súčasne vlastnosti $p_{i_1}, p_{i_2}, ..., p_{i_k}$, potom tento symbol je určený ako mohutnosť zjednotenia množín $A_{i_1} \cap A_{i_2} \cap ... \cap A_{i_k}$

$$N\left(p_{i_1}, p_{i_2}, ..., p_{i_k}\right) = \left|A_{i_1} \cap A_{i_2} \cap ... \cap A_{i_k}\right| \quad (5.23)$$

Duálne tvar tejto formuly dostaneme zámenou p_i za $\overline{p}_i$

$$N\left(\overline{p}_{i_1},\overline{p}_{i_2},...,\overline{p}_{i_k}\right)=\left|\overline{A}_{i_1}\cap\overline{A}_{i_2}\cap...\cap\overline{A}_{i_k}\right|=|U|-\left|A_{i_1}\cup A_{i_2}\cup...\cup A_{i_k}\right| \qquad (5.24)$$

Veta 5.1 (princíp inklúzie a exklúzie).

$$N\left(\overline{p}_1,\overline{p}_2,...,\overline{p}_n\right)=|U|-\sum_{i=1}^{n}N\left(p_i\right)+\sum_{\substack{i,j=1\\(i<j)}}^{n}N\left(p_i,p_j\right)-\sum_{\substack{i,j,k=1\\(i<j<k)}}^{n}N\left(p_i,p_j,p_k\right)+...$$
$$(-1)^n N\left(p_1,p_2,...,p_n\right) \qquad (5.25)$$

Táto veta je priamym dôsledkom formuly (5.20b).

Príklad 5.5. Koľko riešení má rovnica

$$x_1+x_2+x_3=11$$

kde neznáme sú nezáporné celé čísla $x_1\leq 3, x_2\leq 4, x_3\leq 6$. Postulujme tri podmienky, $p_1=\left(x_1>3\right)$, $p_2=\left(x_1>4\right)$ a $p_3=\left(x_1>6\right)$. Použitím vety 5.1 celkový počet riešení je určený formulou

$$N\left(\overline{p}_1,\overline{p}_2,\overline{p}_3\right)=|U|-N\left(p_1\right)-N\left(p_2\right)-N\left(p_3\right)+N\left(p_1,p_2\right)+N\left(p_1,p_3\right)+N\left(p_2,p_3\right)$$
$$-N\left(p_1,p_2,p_3\right) \qquad (\clubsuit)$$

Postupne budeme jednotlivé členy z pravej strany tejto formuly:

(1) $|U|$ = celkový počet nezáporných celočíselných riešení. Nech $0\leq x_1\leq 11$, potom x_2 a x_3 sú určené rovnicou $x_2+x_3=11-x_1=p$, kde $0\leq p\leq 11$. Rovnica $x_2+x_3=p$ má p+1 riešení ($(0,p),(1,p-1),...,(p,0)$). To znamená, že celkový počet riešení rovnice $x_1+x_2+x_3=11$ je

$$|U|=12+11+...+1=\frac{13\cdot 12}{2}=78$$

(2) $N\left(p_1\right)$=počet riešení s $x_1\geq 4$. Použijeme rovnaké argumenty ako v predošlom bode (1), potom celkový počet riešení je

$$N\left(p_1\right)=8+7+...+1=\frac{9\cdot 8}{2}=36$$

(3) $N\left(p_2\right)$=počet riešení s $x_2\geq 5$,potom celkový počet riešení je

$$N\left(p_2\right)=7+6+...+1=\frac{8\cdot 7}{2}=28$$

(4) $N\left(p_3\right)$=počet riešení s $x_3\geq 7$,potom celkový počet riešení je

$$N\left(p_3\right)=5+4+...+1=\frac{6\cdot 5}{2}=15$$

(5) $N\left(p_1,p_2\right)$=počet riešení s $x_1\geq 4$, $x_2\geq 5$, potom celkový počet riešení je určený jednoduchou enumeráciou:A_1=(4,5), A_2= (4,6), A_3= (4,7), A_4= (5,5), A_5= (5,6), A_6= (6,5)

$$N\left(p_1,p_2\right)=6$$

(6) $N\left(p_1,p_3\right)$=počet riešení s $x_1\geq 4$, $x_3\geq 7$, potom celkový počet riešení je určený jednoduchou enumeráciou:A_1=(4,´7)

$$N\left(p_1,p_3\right)=1$$

(7) $N\left(p_2,p_3\right)$=počet riešení s $x_2\geq 5$, $x_3\geq 7$, k týmto podmienkam neexistujú riešenia

$$N(p_2, p_3) = 0$$

(8) $N(p_1, p_2, p_3)$=počet riešení s $x_1 \geq 4$, $x_2 \geq 5$, $x_3 \geq 7$, k týmto podmienkam neexistujú riešenia

$$N(p_1, p_2, p_3) = 0$$

Dosadením týchto partikulárnych výsledkov do (♣) dostaneme požadovaný počat riešený, ktoré $x_1 \leq 3, x_2 \leq 4, x_3 \leq 6$

$$N(\overline{p}_1, \overline{p}_2, \overline{p}_3) = 78 - 36 - 28 - 15 + 6 + 1 + 0 - 0 = 6$$

Eratosthenove sito[6]

Koľko existuje prvočísiel, ktorých hodnota nie je väčšia ako *M*. Pripomeňme, že každé celé číslo *c*, ktoré nie je prvočíslo a ktoré nie je väčšie ako *M*, je vyjadriteľné ako súčin prvočísiel, ktorých kvadrát je menší ako *M*

$$c = p_1^{k_1} p_2^{k_2} p_3^{k_3} .... \leq M \Rightarrow \forall i \left(p_i^2 \leq M \right)$$

kde p_i je *i*-té prvočíslo a exponent k_i je nezáporné celé číslo. Z tejto vlastnosti vychádza aj metóda nazývaná *Eratosthenove sito*. Nech prvočísla, ktoré vyhovujú podmienke $p_i^2 \leq M$ tvoria množinu, obsahujúcu prvých *q* prvočísiel, $\{p_1, p_2, ..., p_k\}$. Množinu prvých *M* kladných celých čísel označíme $C = \{2,3,...,M\}$, kde sme zámerne vynechali 1, pretože 1 priamo z definície nie je prvočíslo. Metódu Eratosthenovho sita vyjadríme pomocou kódu v psudoPascale.

```
C:={2,3,...,M};
for i:=1 to q do
begin for c∈C do
      if (mod(c,pi)=0)∧(c≠pi)then C:=C-{c}
end;
```

Druhý vnútorný cyklus beží nad všetkými elementmi množiny *C*. Celočíselná funkcia *mod*(*a*,*b*) sa nazýva „modulo *a* vzhľadom k *b*“, hodnota funkcie je reprezentovaná zbytkom po delení čísla *a* číslom *b*; ak platí *mod*(*a*,*b*)=0, potom číslo *a* je deliteľné číslom *b*. V prvom kole (*i* = 1, prvé preosievanie Eratosthenovým sitom) algoritmus odstraňuje z množiny *C* všetky čísla, ktoré sú deliteľné $p_i = 2$. Podobne, v druhom kole sa odstraňujú čísla z *C*, ktoré sú deliteľné druhým prvočíslom $p_2 = 3$, atď. Po ukončení „preosievania“ množina *C* obsahuje prvočísla, ktoré nie sú väčšie ako zvolené číslo *M* .

Príklad 5.6. Koľko prvočísiel nie je väčších ako 100?

Použijeme princíp inklúzie a exklúzie (veta 5.1) na riešenie tejto úlohy. Postulujme štyri vlastnosti:

$$P_1 = (\textit{číslo je deliteľné } p_1 = 2)$$

$$P_2 = (\textit{číslo je deliteľné } p_2 = 3)$$

[6] Eratosthenes (276-194 p.n.l.) významný grécky vedec, matematik, geograf, astronóm a filozof, ktorý viedol známu starovekú knižnicu v Alexandrii, svojim súčasníkom bol známy svojou chronológiou starovekej histórie a zmeraním polomeru Zemského globu.

$$P_3 = (\check{c}\acute{\imath}slo\ \ je\ delite\text{ľ}n\acute{e}\ \ p_3 = 5)$$

$$P_4 = (\check{c}\acute{\imath}slo\ \ je\ delite\text{ľ}n\acute{e}\ \ p_4 = 7)$$

Potom počet prvočísiel, ktoré nie sú väčšie ako $M = 100$ je špecifikovaný výrazom

$$4 + N\left(\overline{P}_1, \overline{P}_2, \overline{P}_3, \overline{P}_4\right)$$

Použitím vety 5.1 dostaneme

$$\begin{aligned} N\left(\overline{P}_1, \overline{P}_2, \overline{P}_3, \overline{P}_4\right) = 99 &- N(P_1) - N(P_2) - N(P_3) - N(P_4) + N(P_1, P_2) + N(P_1, P_3) \\ &+ N(P_1, P_4) + N(P_2, P_3) + N(P_2, P_4) + N(P_3, P_4) - N(P_1, P_2, P_3) \\ &- N(P_1, P_2, P_4) - N(P_1, P_3, P_4) - N(P_2, P_3, P_4) + N(P_1, P_2, P_3, P_4) \end{aligned}$$

Veličiny $N(\cdot)$ môžeme jednoducho vyjadriť takto: počet celých čísiel, ktoré nie sú väčšie ako M a ktoré sú deliteľné podmnožinou prvočísel $\{2,3,5,7\}$ je $\lceil 100/x \rceil$, t. j. celou časťou podielu $100/x$, kde x je súčin prvočísiel z danej podmnožiny. Potom

Tabuľka 5.4. Prvočísla menšie ako 100

	2	3_1	4	5_2	6	7_3	8	9	10
11_4	12	13_5	14	15	16	17_6	18	19_7	20
21	22	23_8	24	25	26	27	28	29_9	30
31_{10}	32	33	34	35	36	37_{11}	38	39	40
41_{12}	42	43_{13}	44	45	46	47_{14}	48	49	50
51_{15}	52	53_{16}	54	55	56	57	58	59_{17}	60
61_{18}	62	63	64	65	66	67_{19}	68	69	70
71_{20}	72	73_{21}	74	75	76	77	78	79_{22}	80
81	82	83_{23}	84	85	86	87	88	89_{24}	90
91	92	93	94	95	96	97_{25}	98	99	100

$$\begin{aligned} N\left(\overline{P}_1, \overline{P}_2, \overline{P}_3, \overline{P}_4\right) = 99 &- \left\lceil \frac{100}{2} \right\rceil - \left\lceil \frac{100}{3} \right\rceil - \left\lceil \frac{100}{5} \right\rceil - \left\lceil \frac{100}{7} \right\rceil + \left\lceil \frac{100}{2\cdot 3} \right\rceil + \left\lceil \frac{100}{2\cdot 5} \right\rceil \\ &+ \left\lceil \frac{100}{2\cdot 7} \right\rceil + \left\lceil \frac{100}{3\cdot 5} \right\rceil + \left\lceil \frac{100}{3\cdot 7} \right\rceil + \left\lceil \frac{100}{5\cdot 7} \right\rceil - \left\lceil \frac{100}{2\cdot 3\cdot 5} \right\rceil - \left\lceil \frac{100}{2\cdot 3\cdot 7} \right\rceil \\ &- \left\lceil \frac{100}{2\cdot 5\cdot 7} \right\rceil - \left\lceil \frac{100}{3\cdot 5\cdot 7} \right\rceil + \left\lceil \frac{100}{2\cdot 3\cdot 5\cdot 7} \right\rceil \\ &= 99 - 50 - 33 - 20 - 14 + 16 + 10 + 7 + 6 + 4 + 2 - 3 - 2 - 1 - 0 + 0 \\ &= 21 \end{aligned}$$

To znamená, že existuje 4+21=25 prvočísiel, ktoré nie sú väčšie ako 100, pozri tab. 5.4.

Derangementálne permutácie

Nech P je permutácia n objektov

$$\begin{pmatrix} 1 & 2 & \dots & n \\ r_1 & r_2 & \dots & r_n \end{pmatrix}$$

Permutácia P sa nazýva derangementálna vtedy a len vtedy, ak

$$\forall i \left(i \neq r_i\right)$$

Príkladom derangementálnej permutácie je (2,3,4,5,1) pôvodných 5 obejktov v základnom usporiadaní (1,2,3,4,5). Našim cieľom je spočítať počet D_n derangementálnych permutácií n objektov. Postulujme, že permutácia P má vlastnosť p_i (kde i=1,2,...,n), ak v permutácii v i-tej pozícii je splnená vlastnosť $i = r_i$ (t. j. táto pozícia je zafixovaná). Potom platí

$$D_n = N\left(\overline{p}_1, \overline{p}_2, ..., \overline{p}_n\right)$$

použitím princíp inklúzie a exklúzie (5.25) číslo D_n je určené

$$D_n = n! - \sum_i N\left(p_i\right) + \sum_{i<j} N\left(p_i, p_j\right) - \sum_{i<j<k} N\left(p_i, p_j, p_k\right) + ... \\ + \left(-1\right)^n N\left(p_1, p_2, ..., p_n\right) \qquad (\spadesuit)$$

Určíme jednotlivé veličiny $N(\cdot)$ z pravej strany:

(1) $N\left(p_i\right) = \left(n-1\right)!$

(2) $N\left(p_i, p_j\right) = \left(n-2\right)!$

(3) $N\left(p_i, p_j, p_k\right) = \left(n-3\right)!$

....

Vo všeobecnosti platí

$$N\left(p_{i_1}, p_{i_2}, ..., p_{i_m}\right) = \left(n-m\right)!$$

Musíme ešte pristúpiť k sumácii týchto výrazov

$$\sum_i N\left(p_i\right) = \binom{n}{1}\left(n-1\right)!$$

$$\sum_{i<j} N\left(p_i, p_j\right) = \binom{n}{2}\left(n-2\right)!$$

Vo všeobecnosti platí

$$\sum_{i_1<i_2<...<i_m} N\left(p_{i_1}, p_{i_2}, ..., p_{i_m}\right) = \binom{n}{m}\left(n-m\right)!$$

Ak dosadíme tieto formuly do výrazu ($\spadesuit$) pre počet derangementálnych permutácií, po jednoduchých úpravách dostaneme konečný výraz pre D_n.

Veta 5.2. Počet derangementálnych permutácií n objektov je určený

$$D_n = n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + ... + \left(-1\right)^n \frac{1}{n!}\right) \qquad (5.26)$$

V rôznych aplikáciach je zaujimavá otázka, s akou pravdepodobnosťou pri náhodnej generácii permutácie n objektov dostaneme derangementálnu permutáciu. Táto pravdepodobnosť je určená pomerom $D_n/n!$, použitím (5.26) dostaneme jendnoduchú formulu pre túto pravdepodobmnosť

$$\frac{D_n}{n!} = \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + ... + \left(-1\right)^n \frac{1}{n!}\right) \xrightarrow{n\to\infty} e^{-1} \qquad (5.27)$$

Poznámka. Kartová hra *recontres* bola vo Franci 17.-18. storočia veľmi populárna. Spočívala v tom, že 52 kariet (z ktorých bolo 26 párov rovnakých kariet) sa rozdalo a usporiadalo do dvoch radov pod sebou tak, že vždy dvojica kariet tvorila pár. Skóre hry sa počítalo podľa množstva rovnakých párov. Preto v r. 1708 francúzsky matematik Pierre Raymond de Montmort(1678-1719) riešil úlohu, s akou pravde-podobnosť dostane hráč také karty, že neexistuje priradenie medzi dvojicami kariet, ako vyplýva z (5.27)′, táto pravdepodobnosť je približne rovná $1/e \approx\sim 0.368$. Táto pomerne malá pravdepodobnosť (okolo 40%) robila túto hru zaujímavou v určitých kruhoch vtedajšej aristokracie.

Cvičenia

Cvičenie 5.1. Nájdite prvých päť členov postupnosti, z ktorých každá je definovaná rekuretnou formulou a počiatočnou podmienkou:
(a) $a_n = 6a_{n-1}$, $a_0 = 2$,
(b) $a_n = a_{n-1}^2$, $a_0 = 2$,
(c) $a_n = a_{n-1} + 3a_{n-2}$, $a_0 = 1, a_1 = 2$,
(d) $a_n = na_{n-1} + n^2 a_{n-2}$, $a_0 = 1, a_1 = 1$,
(e) $a_n = a_{n-1} + a_{n-3}$, $a_0 = 1, a_1 = 2, a_2 = 0$.

Cvičenie 5.2. Nech $a_n = 2^n + 5 \cdot 3^n$, pre n = 0, 1, 2, ...
(a) Nájdite a_0, a_1, a_2, a_3, a_4,
(b) ukážte, že $a_2 = 5a_1 - 6a_0$, $a_3 = 5a_2 - 6a_1$ a $a_4 = 5a_3 - 6a_2$.
(c) ukážte, že $a_n = 5a_{n-1} - 6a_{n-2}$, pre $n \geq 2$.

Cvičenie 5.3. Ukážte, že postupnosť $\{a_n\}$ je riešením rekuretnej formuly $a_n = -3a_{n-1} + 4a_{n-2}$ ak
(a) $a_n = 0$,
(b) $a_n = 1$,
(c) $a_n = (-4)^n$,
(d) $a_n = 2(-4)^n + 3$.

Cvičenie 5.4. Do banky ste uložili 1000 EUR na 3% ročný úrok.
(a) Zostrojte rekurentnú formulu pre výpočet veľkosti vkladu,
(b) zostrojte explicitnú formulu na veľkosť vkladu po *n* rokoch,
(c) aká bude veľkosť vkladu po 100 rokoch.

Cvičenie 5.5. Akú sumu v EUR musíte vložiť do banky na 3% ročný úrok, aby ste si po dobu 15 rokov mohli mesačne vyberať z banky 1000 EUR tak, aby po to 15 rokoch vklad bol úplne vybraný.

Cvičenie 5.6. Koľko existuje binárnych reťazcov dĺžky 7, ktoré obsahujú párny počet núl (t. j. 0, 2, 4 a 6)?

Cvičenie 5.7. V rámci metódy „rozdeľuj-a-panuj" bola odvodená rekurentná formula $f(n) = 5f(n/2) + 3$, s počiatočnou podmienkou $f(1) = 7$. Nájdite $f(2^k)$, kde *k* je kladné celé číslo. Taktiež vyhodnoťte $f(n)$, pre $n \to \infty$.

Cvičenie 5.8. Koľko porovnaní je potrebných pre binárne prehľadávanie v postupnosti, ktorá obsahuje 64 elementov.

Cvičenie 5.9. Predpokladajme, že $f(n) = f(n/3) + 1$, pričom n je deliteľné 3 a $f(1)$=1. Nájdite

(a) $f(3)$,
(b) $f(27)$,
(c) $f(729)$.

Cvičenie 5.10. Predpokladajme, že $f(n) = 2f(n/2) + 3$, pričom n je párne $f(1)=5$. Nájdite
(a) $f(2)$,
(b) $f(8)$,
(c) $f(64)$,
(d) $f(1024)$.

Cvičenie 5.11. Nájdite $f(n)$, kde $n = 2^k$, funkcia f je určená rekurentnou reláciou $f(n) = f(n/2) + 1$, $f(1)=1$. Ako sa správa funkcia f pre $n \to \infty$.

Cvičenie 5.12. Nech na turnaji je 2^k druzštiev, turnaj prebieha eliminačným spôsobom, t. j. do ďalšieho kola postupuje len víťazi z predchádzajúceho kola. Zostrojte rekurenčnú reláciu pre počet vzájomných zápasov v turnaji.

Cvičenie 5.13. Nech funkcia f vyhovuje rekurečnej relácii $f(n) = 2f(\sqrt{n}) + 1$, kde n je väčšie ako 1 a je odmnocniteľné, počiatočná podmienka má tvar f(2)=1. Nájdite
(a) $f(16)$,
(b) asymptotickú formulu pre $n \to \infty$.

Cvičenie 5.14. Koľko elementov obsahuje prienik $A_1 \cup A_2$, ak $|A_1| = 12$, $|A_2| = 18$,
(a) $A_1 \cap A_2 = \varnothing$,
(b) $|A_1 \cap A_2| = 1$.,
(c) $|A_1 \cap A_2| = 6$,
(d) $A_1 \subseteq A_2$.

Cvičenie 5.15. Na fakulte je 345 študentov, ktorí si zapísali predmet Matematická analýza, 212 študentov, ktorí si zapísali predmet Diskrétna matematika a 188 študentov, ktorí si zapísali súčasne predmety Matematická analýza a Diskrétna matematika. Koľko študentov má zapísaný len predmet Matematická analýza alebo predmet Diskrétna matematika.

Cvičenie 5.16. Koľko elementov obsahuje zjednotenie $A_1 \cup A_2 \cup A_3$, ak každá množina obsahuje 100 elementov, pričom
(a) množiny sú po dvojiciach disjunktné,
(b) každá dvojica množín obsahuje 50 spoločných elementov a žiadny element, ktorý by sa súčasne nachádzal vo všetkých troch množinách,
(c) každá dvojica množín obsahuje 50 spoločných elementov a 25 eleemntov, ktoré sa nachádzajú súčasne vo všetkých troch množinách.

Cvičenie 5.17. Nájdite počet kladných celých čísel, ktoré nie sú väčšie ako 100 a ktoré nie sú deliteľné 5 alebo 7.

Cvičenie 5.18. Nájdite počet kladných celých čísel, ktoré nie sú väčšie ako 100 a ktoré sú párne alebo sú kvadrátom nejakého celého čísla.

Cvičenie 5.19. Koľko binárnych reťazcov dĺžky 8 neobsahuje podreťazec ´000000´.

Cvičenie 5.20. V koši máme 100 jabĺčok, z ktorých 20 červivých a 15 je nahnitých. Nech v koši je 10 jabĺčok, ktoré sú červivé a nahnité, koľko jabĺčok v koši nie je ani červivých a ani nahnitých?

Cvičenie 5.21. Koľko nezáporných celočíselných riešení menších ako 6 má rovnica $x_1 + x_2 + x_3 = 13$?

Cvičenie 5.22. Napíšte všetky derangementy 4 objektov.