Kombinatorik

Douglas Wikström
KTH Stockholm
popup-help@csc.kth.se

Binomialtal och Pascals triangel (1/3)

$\binom{n}{k}$ är antalet sätt att välja en k-delmängd från en mängd med n element. Vi summerar antalet sätt att välja k element där första elementet är med respektive inte med.

$$\binom{n}{k} = \begin{cases} 1 & \text{om } n = k \text{ eller } k = 0 \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{annars} \end{cases}$$

Pascals triangel:

												k
						1						0
					1		1					1
				1		2		1				2
			1		3		3		1			3
		1		4		6		4		1		4
	1		5		10		10		5		1	5

Binomialtal och Pascals triangel (2/3)

Dynamisk programmering med memoisering. $T[i, j]$ är en tabell där alla element från början är satta till $\perp$. Tabellen kan återanvändas mellan anrop.

Input: Heltal n och k, $0 \leq k \leq n$.
Output: $\binom{n}{k}$

$\textsc{Bin}(n, k)$
(1) **if** $T[n, k] = \perp$
(2) **if** $n = k \vee k = 0$
(3) $T[n, k] = 1$
(4) **else**
(5) $T[n, k] \leftarrow \textsc{Bin}(n-1, k-1) + \textsc{Bin}(n-1, k)$
(6) **return** $T[n, k]$

Binomialtal och Pascals triangel (3/3)

Vanlig dynamisk programmering från botten och upp.

Input: Heltal n och k, $0 \leq k \leq n$.
Output: $\binom{n}{k}$

$\textsc{Bin}(n, k)$

```
for i = 0 to n
    T[i, 0] ← T[i, i] ← 1
for i = 2 to n
    for j = 1 to min{i − 1, k}
        T[i, j] ← T[i − 1, j − 1] + T[i − 1, j]
return T[n, k]
```

Multinomialtal

Antag att vi har t lådor av storlekarna $k_1, \ldots, k_t$, och vill veta hur många sätt vi kan placera elementen $1, \ldots, n$ i lådorna, där $n = \sum_{i=1}^{t} k_i$. Multinomialtal betecknar detta antal och kan definieras:

$$\binom{n}{k_1, \ldots, k_t} = \frac{n!}{k_1! \cdot \ldots \cdot k_t!}$$

Mer koncist beskrivs detta av följande:

$$\left(\sum_{i=1}^{t} x_i\right)^n = \sum_{\sum_{i=1}^{t} k_i = n} \binom{n}{k_1, \ldots, k_t} \prod_{i=1}^{t} x_i^{k_i}$$

Multinomialtal

För att beräkna multinomialtal kan man utnyttja sambandet:

$$\binom{n}{k_1, \ldots, k_t} = \binom{k_1}{k_1}\binom{k_1 + k_2}{k_2} \cdot \ldots \cdot \binom{\sum_{j=1}^{t} k_j}{k_i}$$

Man kan även formulera en slags generaliserad Pascals triangel, men det är krångligt.

Generera permutationer

Givet en sträng $x_1 x_2 \ldots x_k$, möjligen med flera kopior av några tecken, hur genererar vi alla permutationer av dess symboler i lexikografisk ordning?

Generera permutationer

$\textsc{Next}(x_1 x_2 \ldots x_n)$

(1) Hitta i så att $x_i < x_{i+1} \geq \ldots \geq x_n$.

(2) Hitta största j så att $x_j > x_i$.

(3) $\textsc{Swap}(x_i, x_j)$

(4) $\textsc{Reverse}(x_{i+1} \ldots x_n)$

(5) **return** $x_1 x_2 \ldots x_n$

Generera alla k-delmängder

Givet en mängd $\{1, \ldots, n\}$, hur genererar vi alla dess k-delmängder?

Vi kan representera en delmängd S med en binär sträng $x_1 x_2 \ldots x_n$, där $x_i = 1$ omm $i \in S$.

Generera alla k-delmängder

Alternativ:

1. Generera alla delmängder och filtrera.
2. Generera med NEXT.
3. Bitpillande implementation av NEXT, dvs BINARYNEXT (förutsätter tvåkomplementsrepresentation).

BINARYNEXT$(x_1 x_2 \ldots x_n)$

(1) $t \leftarrow x \wedge (-x)$

(2) $r \leftarrow x + t$

(3) **return** $((r \oplus x)/(4t)) \vee r$

Partitionering av en mängd

Definition. En mängd $\{B_1, \ldots, B_k\}$ är en partitionering av mängden $[n] = \{1, \ldots, n\}$ om och endast om:

1. $\bigcup_i B_i = [n]$,
2. $B_i \cap B_j = \emptyset$, och
3. $B_i \neq \emptyset$.

Hur många partitioner av $[n]$ i k block finns det?

Stirlingtal (av andra sorten)

$$S(n,0) = 0 \text{ om } n > 0$$
$$S(n,1) = 1$$
$$S(n,n) = 1$$
$$S(n,k) = S(n-1,k-1) + k \cdot S(n-1,k)$$

Belltal (totalt antal partitioneringar)

Belltal beskriver totala antalet sätt att partitionera en mängd, och uppfyller följande samband:

$$B(n) = \sum_{i=0}^{n} S(n,i)B(n+1) \qquad = \sum_{i=0}^{n} \binom{n}{n-i} B(i)$$

Catalantal

Vad är antalet ordnade binära träd med n noder?

Catalantal

Antalet ordnade träd med n noder ges av:

$$C(n) = \sum_{k=0}^{n-1} C(k)C(n-1-k)$$

Men det är bättre att använda:

$$C(n) = \frac{\binom{2n}{n}}{n+1}$$