To construct a ranking algorithm, we need to count the number of k-element subsets preceding a given set T in this ordering. Suppose that t_1 is an integer such that $1 \le t_1 \le n$. It is easy to see that there are exactly $\binom{n-t_1}{k-1}$ subsets $X \in \mathcal{S}$ such that $x_1 = t_1$, where $\vec{X} = [x_1, \ldots, x_k]$. More generally, for any $i \le k$ integers $t_1, \ldots, t_i$ such that $1 \le t_1 < \cdots < t_i \le n$, there are exactly $\binom{n-t_i}{k-i}$ subsets $X \in \mathcal{S}$ such that $x_1 = t_1, \ldots$, and $x_i = t_i$.

Now, suppose that $T \in \mathcal{S}$, and $\vec{T} = [t_1, t_2, \ldots, t_k]$ is defined as above. The k-element subsets X preceding T in lexicographic order are the following:

- The subsets X with $1 \le x_1 \le t_1 - 1$.
- The subsets X with $x_1 = t_1$ and $t_1 + 1 \le x_2 \le t_2 - 1$.
- The subsets X with $x_1 = t_1, x_2 = t_2$, and $t_2 + 1 \le x_3 \le t_3 - 1$.
- etc.
- The subsets X with $x_1 = t_1, x_2 = t_2, \ldots, x_{k-1} = t_{k-1}$ and $t_{k-1} + 1 \le x_k \le t_k - 1$.

From these facts, we can write down a formula for $\mathsf{rank}(T)$, where $\vec{T} = [t_1, t_2, \ldots, t_k]$. We get the following formula, where we define $t_0 = 0$ for convenience:

$$\mathsf{rank}(T) = \sum_{i=1}^{k} \sum_{j=t_{i-1}+1}^{t_i - 1} \binom{n-j}{k-i}.$$

This formula immediately yields a ranking algorithm, which we present as Algorithm 2.7.

Algorithm 2.7: KSUBSETLEXRANK $(\vec{T}, k, n)$

```
r ← 0
t_0 ← 0
for i ← 1 to k
    do { if t_{i-1} + 1 ≤ t_i - 1
         then { for j ← t_{i-1} + 1 to t_i - 1
                  do r ← r + C(n-j, k-i)
return (r)
```

Now we unravel Algorithm 2.7 to obtain an unranking algorithm. Suppose that $0 \le r \le \binom{n}{k} - 1$, and suppose that $T = \mathsf{unrank}(r)$ with $\vec{T} = [t_1, \ldots, t_k]$. The smallest element in T, t_1, can be determined by the observation that

$$t_1 = x \Leftrightarrow \sum_{j=1}^{x-1} \binom{n-j}{k-1} \le r < \sum_{j=1}^{x} \binom{n-j}{k-1}.$$

Having determined t_1, we can compute t_2 in a similar way:

$$t_2 = x \Leftrightarrow \sum_{j=t_1+1}^{x-1} \binom{n-j}{k-2} \le r - \sum_{j=1}^{t_1-1} \binom{n-j}{k-1} < \sum_{j=t_1+1}^{x} \binom{n-j}{k-2}.$$

The pattern continues, and the entire algorithm is presented as Algorithm 2.8.

Algorithm 2.8: KSUBSETLEXUNRANK (r, k, n)

```
x ← 1
for i ← 1 to k
    do { while C(n-x, k-i) ≤ r
           do { r ← r - C(n-x, k-i)
                x ← x + 1
         t_i ← x
         x ← x + 1
return (T⃗)
```

2.3.2 Co-lex ordering

There is a useful alternative to the lexicographic ordering for k-element subsets of an n-set. The ordering is called the *co-lex ordering*, and it is defined as follows. A k-element subset $T \subseteq S$ is written as a list

$$\overleftarrow{T} = [t_1, t_2, \ldots, t_k],$$

where

$$t_1 > t_2 > \cdots > t_k.$$

The co-lex ordering is induced by the lexicographic ordering on the sequences $\overleftarrow{T}$ $(T \in \mathcal{S})$.

We illustrate the co-lex ordering when $n = 5$ and $k = 3$. The co-lex ordering of the ten 3-element subsets of $\{1, \ldots, 5\}$ is as follows:

T	$\overleftarrow{T}$	$\mathsf{rank}(T)$
$\{1,2,3\}$	$[3,2,1]$	0
$\{1,2,4\}$	$[4,2,1]$	1
$\{1,3,4\}$	$[4,3,1]$	2
$\{2,3,4\}$	$[4,3,2]$	3
$\{1,2,5\}$	$[5,2,1]$	4
$\{1,3,5\}$	$[5,3,1]$	5
$\{2,3,5\}$	$[5,3,2]$	6
$\{1,4,5\}$	$[5,4,1]$	7
$\{2,4,5\}$	$[5,4,2]$	8