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 $1 \leq t_1 \leq n$. It is easy to see that there are exactly $\binom{n-1}{k-1}$ subsets $X \in S$ such that $x_1 = t_1$, where $X = \{x_1, \ldots, x_k\}$. More generally, for any $i \leq k$ integers $t_1, \ldots, t_i$ such that $1 \leq t_1 < \cdots < t_i \leq n$, there are exactly $\binom{n-i}{k-i}$ subsets $X \in S$ such that $x_1 = t_1, \ldots, x_i = t_i$.

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

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

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

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

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

**Algorithm 2.7: kSUBSETLEXRANK $(\overline{T}, k, n)$**

```plaintext
r ← 0
t₀ ← 0
for i ← 1 to k
    do if $t_{i-1} + 1 \leq t_i - 1$
        then for j ← $t_{i-1} + 1$ to $t_i - 1$
            do r ← r + $\binom{n-j}{k-i}$
    return (r)
```

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

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

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

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

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

**Algorithm 2.8: kSUBSETLEXUNRANK $(r, k, n)$**

```plaintext
x ← 1
for i ← 1 to k
    do while $\binom{n-i}{k-i} \leq r$
        do $r \leftarrow r - \binom{n-x}{k-x}$
        do $x \leftarrow x + 1$
        tᵢ ← x
        x ← x + 1
return (T)
```

### 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 $\overline{T} = [t_1, t_2, \ldots, t_k]$, where

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

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

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:

<table>
<thead>
<tr>
<th>$T$</th>
<th>$\overline{T}$</th>
<th>rank($T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1, 2, 3}$</td>
<td>$[3, 2, 1]$</td>
<td>0</td>
</tr>
<tr>
<td>${1, 2, 4}$</td>
<td>$[4, 2, 1]$</td>
<td>1</td>
</tr>
<tr>
<td>${1, 3, 4}$</td>
<td>$[4, 3, 1]$</td>
<td>2</td>
</tr>
<tr>
<td>${2, 3, 4}$</td>
<td>$[4, 3, 2]$</td>
<td>3</td>
</tr>
<tr>
<td>${1, 2, 5}$</td>
<td>$[5, 2, 1]$</td>
<td>4</td>
</tr>
<tr>
<td>${1, 3, 5}$</td>
<td>$[5, 3, 1]$</td>
<td>5</td>
</tr>
<tr>
<td>${2, 3, 5}$</td>
<td>$[5, 3, 2]$</td>
<td>6</td>
</tr>
<tr>
<td>${1, 4, 5}$</td>
<td>$[5, 4, 1]$</td>
<td>7</td>
</tr>
<tr>
<td>${2, 4, 5}$</td>
<td>$[5, 4, 2]$</td>
<td>8</td>
</tr>
</tbody>
</table>