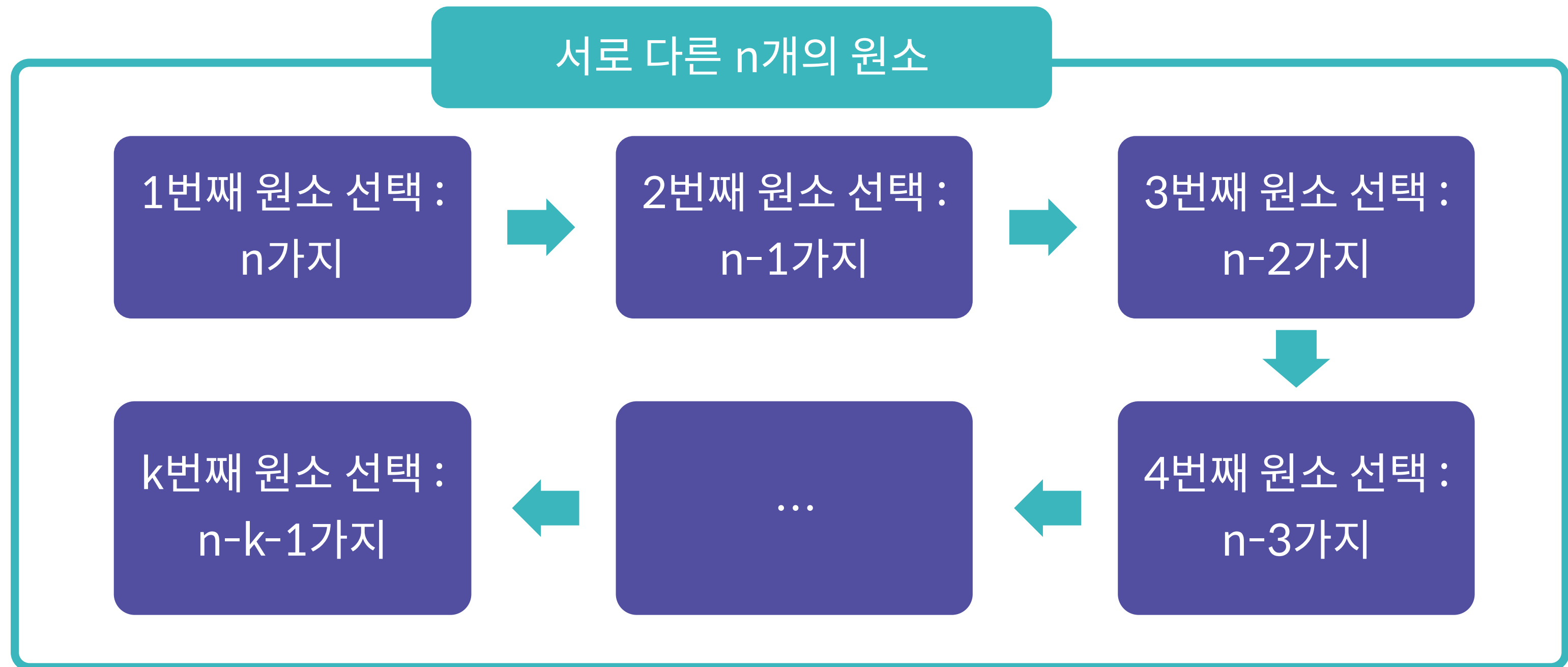


순열과 조합

순열



곱의 법칙에 의해 총 가능한 경우의 수
= n 개의 서로 다른 원소 중 k 개를 선택하여 배열하는 경우의 수 = 순열

$$n(n-1)\cdots(n-k+1) = {}_n P_k$$

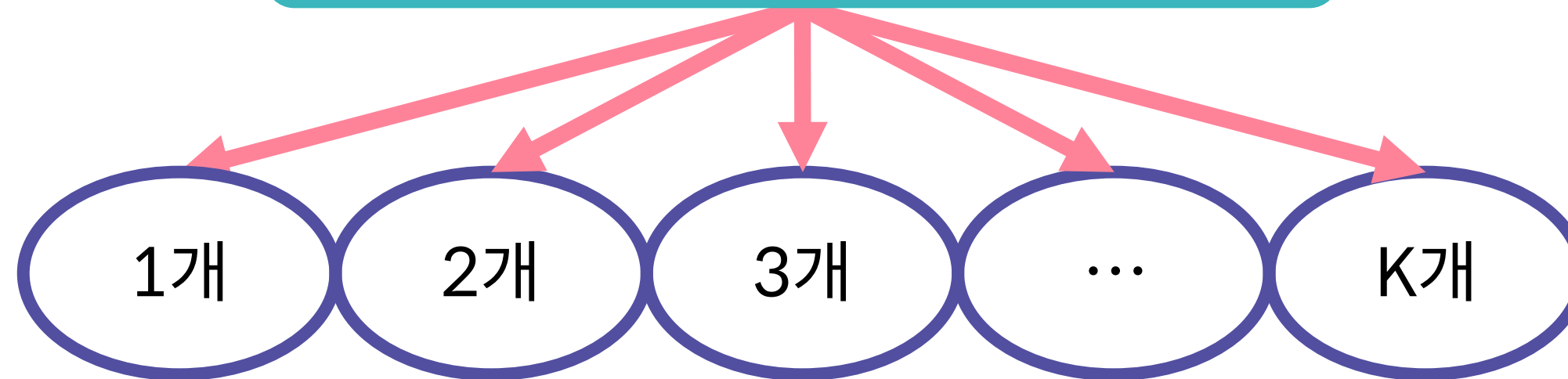
순열

```
from itertools import permutations  
  
list(permutations([n], k))
```

[] 안에 서로 다른 n개의 원소를 주고, 이 원소들 중 k개를 순서를 고려하여 뽑는 경우의 수 계산

조합

서로 다른 n 개의 원소



서로 다른 n 개의 원소에서 k 개를 순서에 상관없이 선택하는 방법

$$\begin{aligned} &= \frac{\text{(n개의 서로 다른 원소 중 k개를 선택하여 배열하는 경우의 수)}}{\text{(K개의 원소를 나열하는 경우의 수)}} = \frac{n(n-1) \dots (n-k+1)}{k!} \end{aligned}$$

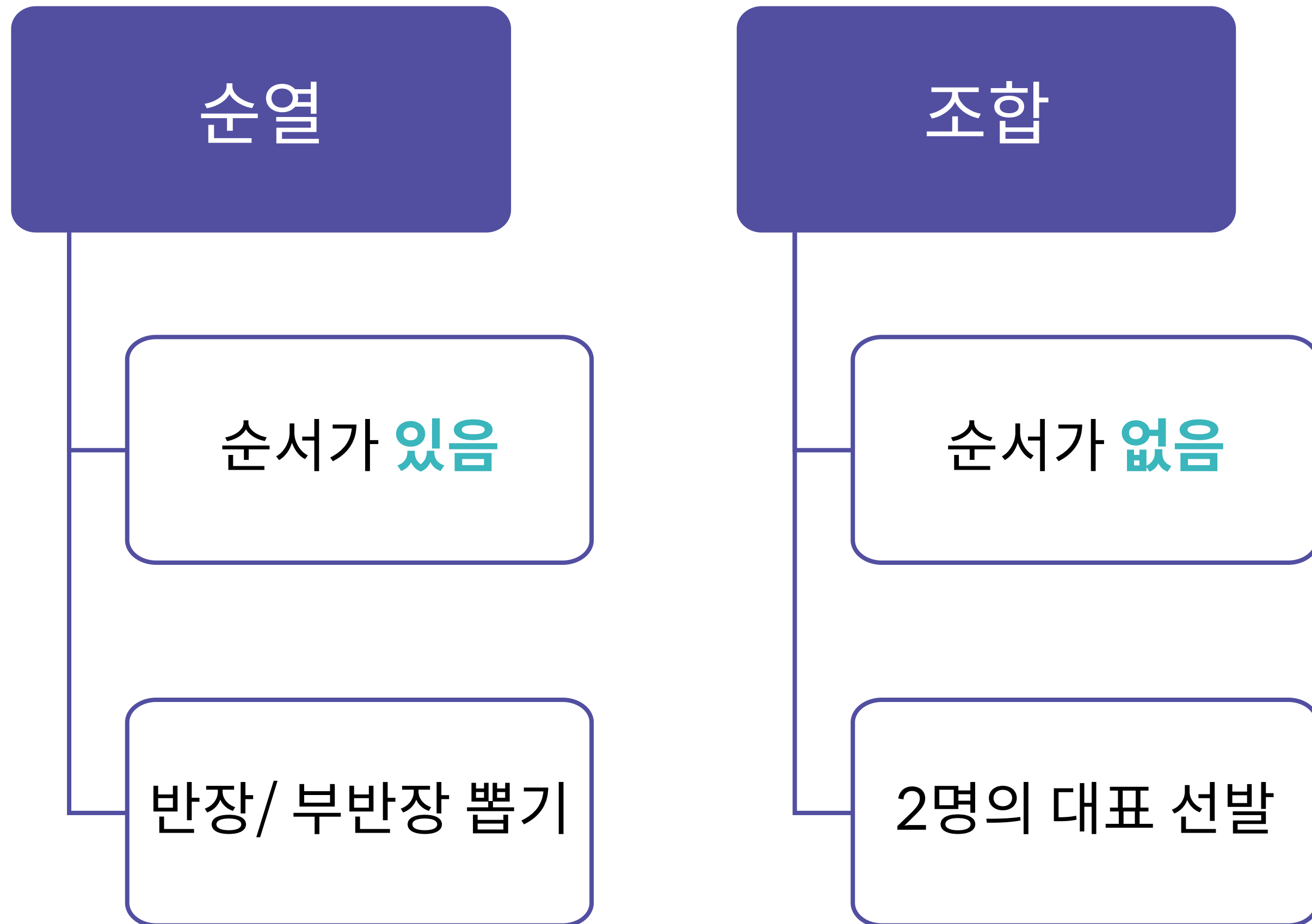
$$= {}_n C_k$$

조합

```
from itertools import combinations  
list(combinations([n], k))
```

[] 안에 서로 다른 n개의 원소를 주고, 이 원소들 중 k개를
순서를 고려하지 않고 뽑는 경우의 수 계산

순열과 조합의 차이



중복순열

서로 다른 n 개의 원소 중에서 중복을 허용하여
 r 개를 뽑아 일렬로 배열하는 경우

$$nPr = n^r$$

예) a, b 중에서 중복을 허용하여 세 개를 뽑아 배열하는 경우 = 8개

a	a	a	a	b	a	b	a	a	b	b	a
a	a	b	a	b	b	b	a	b	b	b	b

중복순열

```
from itertools import product  
list(product([n], repeat = k))
```

[] 안에 서로 다른 n개의 원소를 주고, 이 원소들 중 k개를 중복을 허용하면서 순서를 고려하여 뽑는 경우의 수 계산

중복조합

서로 다른 n 개의 대상 중 중복을 허용해
 r 개를 순서를 고려하지 않고 뽑는 경우

$${}_nH_r = {}_{n+r-1}C_r$$

예) 집합 $S = \{1,2,3,4\}$ 에서
중복을 허용하여 3개의 원소를 뽑는 경우
 $(1,1,1), (1,1,2), (1,1,3), \dots, (4,4,4)$

$${}_4H_3 = {}_{4+3-1}C_3 = {}_6C_3 = 20$$

중복조합

```
from itertools import combinations
list(combinations_with_replacement([n], k))
```

[] 안에 서로 다른 n개의 원소를 주고, 이 원소들 중 k개를
순서를 중복을 허락하여 순서를 고려하지 않고 뽑는
경우의 수 계산