

- 순열

- $P(n,n) = n * (n-1) !$

1 2 3 4	2 1 3 4	3 1 2 4	4 1 2 3
1 2 4 3	2 1 4 3	3 1 4 2	4 1 3 2
1 3 2 4	2 3 1 4	3 2 1 4	4 2 1 3
1 3 4 2	2 3 4 1	3 2 4 1	4 2 3 1
1 4 2 3	2 4 1 3	3 4 1 2	4 3 1 2
1 4 3 2	2 4 3 1	3 4 2 1	4 3 2 1



1로 끝나는 수열

2 3 4 1  
 2 4 3 1  
 3 2 4 1  
 3 4 2 1  
 4 2 3 1  
 4 3 2 1

2로 끝나는 수열

1 3 4 2  
 1 4 3 2  
 3 1 4 2  
 3 4 1 2  
 4 1 3 2  
 4 3 1 2

3로 끝나는 수열

1 2 4 3  
 1 4 2 3  
 2 1 4 3  
 2 4 1 3  
 4 1 2 3  
 4 2 1 3

4로 끝나는 수열

1 2 3 4  
 1 3 2 4  
 2 1 3 4  
 2 3 1 4  
 3 1 2 4  
 3 2 1 4

- 순열 생성 재귀적 알고리즘1

```
perm(n, r)
    if (r == 0) print_arr()
    else
        for (i : n - 1 ~ 0)
            swap(a[i], a[n - 1])
            t[r - 1] = a[n - 1]
            perm(n - 1, r - 1)
            swap(a[i], a[n - 1])
```

- 순열 생성 재귀적 알고리즘2

```
perm(k)
    if (k == R) print_arr()
    else
        for (i : k ~ N - 1)
            swap(k, i)
            perm(k + 1)
            swap(k, i)
```

- 순열 생성 재귀적 알고리즘2

```
Visited[N-1]
perm(k)
    if (k == N) print_arr()
    else
        for (i : 0 ~ N - 1)
            if (visited[i]) continue
            t[k] = a[i]
            visited[i] = true
            perm(k + 1)
            visited[i] = false
```

# 조합

 서로 다른  $n$ 개의 원소 중  $r$ 개를 순서 없이 골라낸 것을 조합

$${}_nC_r = \frac{n!}{(n-r)!r!}, (n \geq r)$$

$${}_nC_r = {}_{n-1}C_{r-1} + {}_{n-1}C_r \longrightarrow \text{재귀적 표현}$$

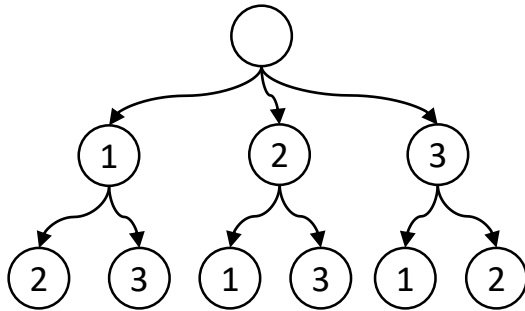
$${}_nC_0 = 1$$

## 재귀 호출을 이용한 조합 생성 알고리즘

```
comb(n, r)
    IF r == 0 : print_array_t()
    ELIF n < r : RETURN
    ELSE
        tr[r - 1] ← an[n - 1]
        comb(n - 1, r - 1)
        comb(n - 1, r)
```

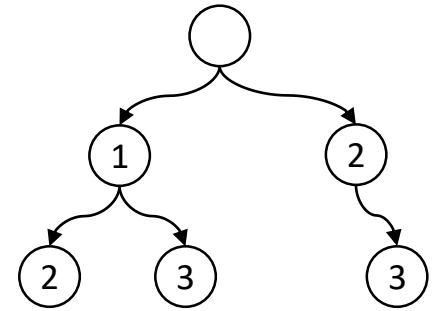
$N = 3, R = 2$

1 2  
1 3  
~~2 1~~  
2 3  
~~3 1~~  
~~3 2~~



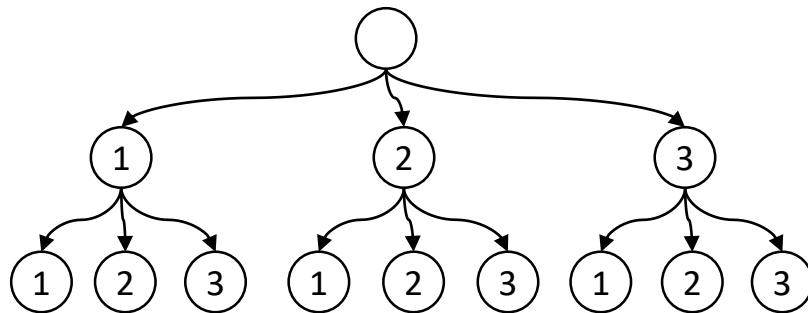
순열

1 2  
1 3  
2 3



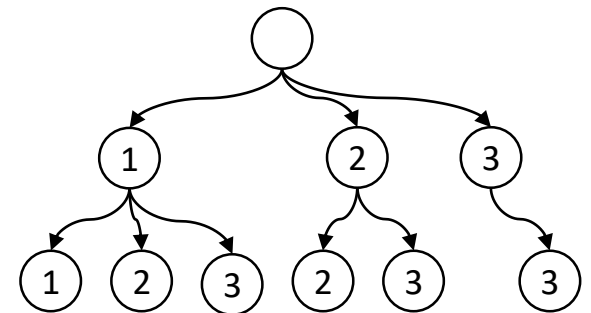
조합

1 1  
1 2  
1 3  
~~2 1~~  
2 2  
2 3  
2 3  
~~3 1~~  
~~3 2~~  
3 3



중복 순열

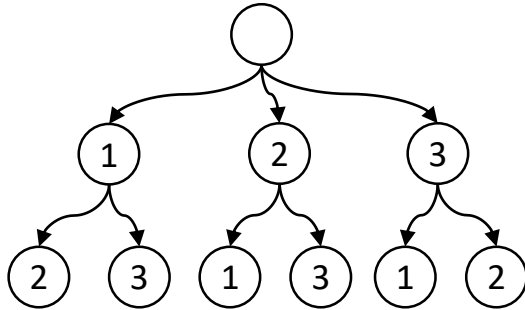
1 1  
1 2  
1 3  
2 2  
2 3  
2 3  
3 3



중복 조합

$N = 3, R = 2$

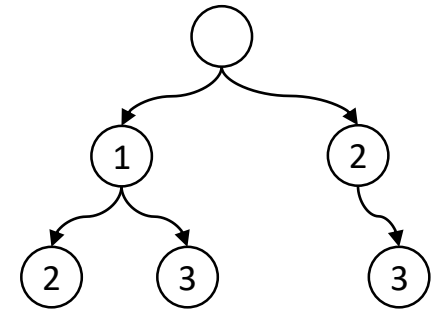
1 2  
1 3  
~~2 1~~  
2 3  
~~3 1~~  
~~3 2~~



```

for x1 : 1 ~ 3
  for x2 : 1 ~ 3 단, x2 != x1
    print( x1, x2 )
  
```

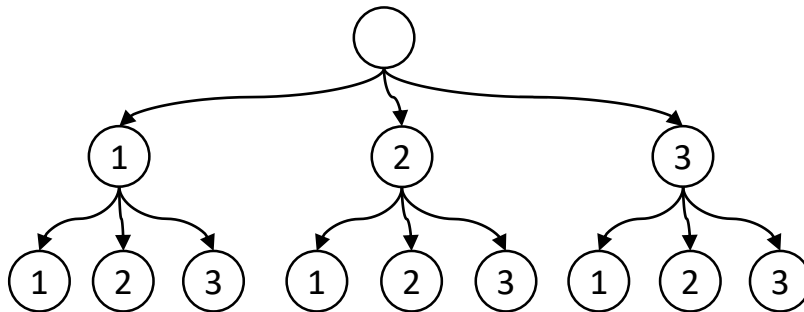
1 2  
1 3  
2 3



```

for x1 : 1 ~ 2
  for x2 : 2 ~ 3
    print( x1, x2 )
  
```

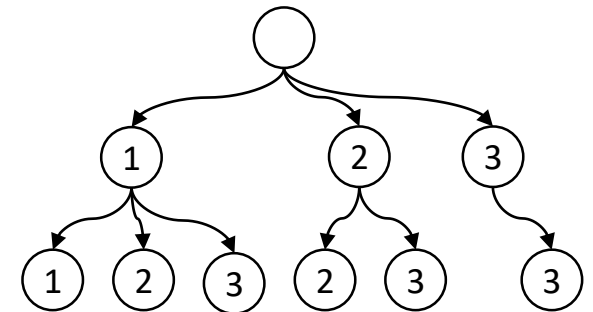
1 1  
1 2  
1 3  
~~2 1~~  
2 2  
2 3  
~~3 1~~  
~~3 2~~  
3 3



```

for x1 : 1 ~ 3
  for x2 : 1 ~ 3
    print( x1, x2 )
  
```

1 1  
1 2  
1 3  
2 2  
2 3  
3 3



```

for x1 : 1 ~ 3
  for x2 : x1 ~ 3
    print( x1, x2 )
  
```



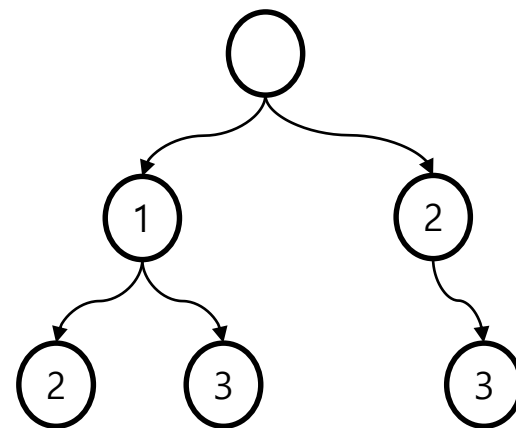
- 조합 생성 재귀적 알고리즘1

$$C(n, r) = C(n - 1, r - 1) + C(n - 1, r)$$

```
comb(n, r)
    if( r == 0) print_arr()
    else if (n < r) return
    else
        t[r-1] = a[n-1]
        comb(n-1, r-1)
        comb(n-1, r)
```

- 조합 생성 재귀적 알고리즘2

- 초기값 :  $k = 0, s = 0, N, R$



```
comb(k, s) // 깊이, 시작숫자
    if (k == R) print_arr()
    else
        for (int i : s ~ N - R + k )
            t[k] = a[i]
            comb(k + 1, i + 1)
```

- 중복 순열 생성 재귀적 알고리즘1

```
PI(n, r)
```

```
    if (r == 0) print()
```

```
    else
```

```
        for (i : n - 1 ~ 0)
```

```
            swap(a[i], a[n - 1])
```

```
            t[r - 1] = a[n - 1]
```

```
            PI(n, r - 1)
```

```
            swap(a[i], a[n - 1])
```

```
perm(n, r)
```

```
    if (r == 0) print()
```

```
    else
```

```
        for (i : n - 1 ~ 0)
```

```
            swap(a[i], a[n - 1])
```

```
            t[r - 1] = a[n - 1]
```

```
            perm(n - 1, r - 1)
```

```
            swap(a[i], a[n - 1])
```

- 중복 순열 생성 재귀적 알고리즘2  $n^r$

```
PI(k) // 깊이
```

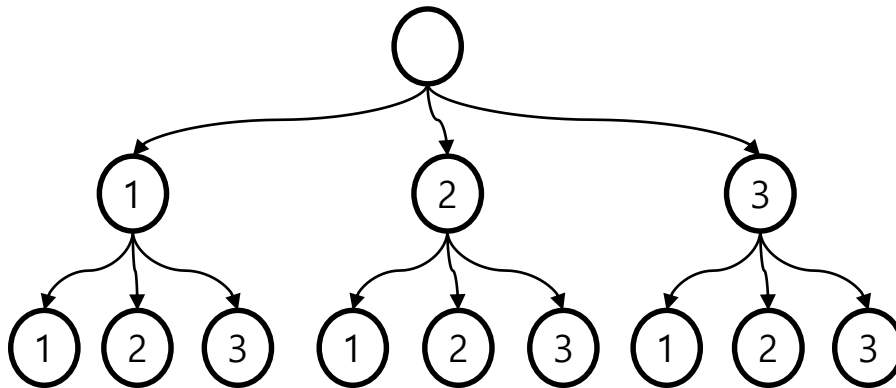
```
    if (k == R) print_arr()
```

```
    else
```

```
        for (i : 0 ~ N - 1)
```

```
            t[k] = a[i]
```

```
            pi_r(k + 1)
```



- 중복조합 생성 재귀적 알고리즘1

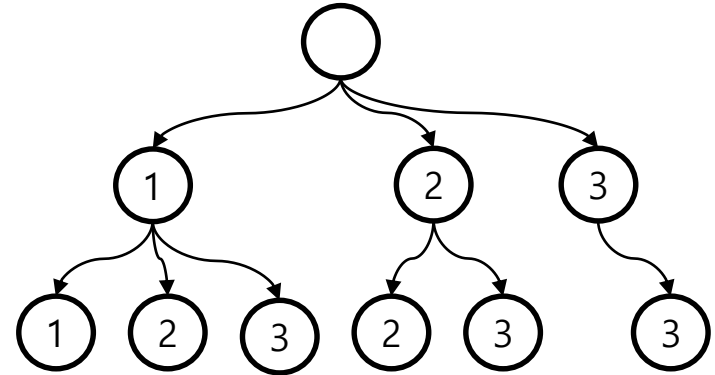
```
H(n, r)
    if( r == 0) print_arr()
    else if (n == 0) return
    else
        t[r-1] = a[n-1]
        H(n, r-1)
        H(n-1, r)
```

```
comb(n, r)
    if( r == 0) print_arr()
    else if (n < r) return
    else
        t[r-1] = a[n-1]
        comb(n-1, r-1)
        comb(n-1, r)
```

$$H(n, r) = H(n, n-1) + H(n-1, r)$$

- **중복조합 생성 재귀적 알고리즘2**

- 초기값 :  $K = 0, s = 1, N, R$



```
H(k, s) // 깊이, 시작숫자
    if (k == R) print_arr()
    else
        for (int i : s ~ N )
            t[k] = a[i]
            H(k + 1, i)
```