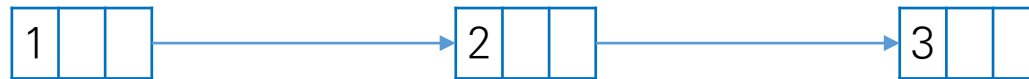


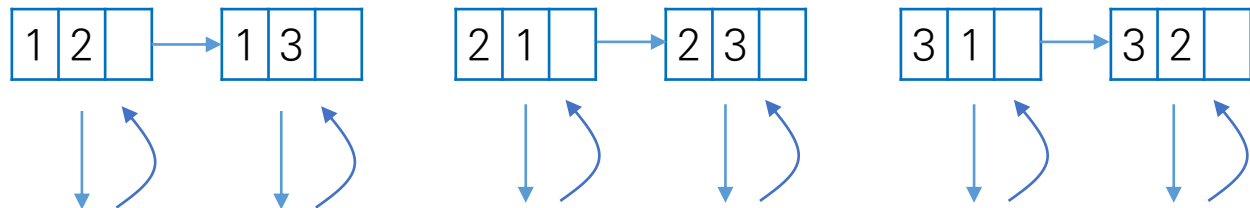
순열

■ 1, 2, 3으로 3자리 수 만들기

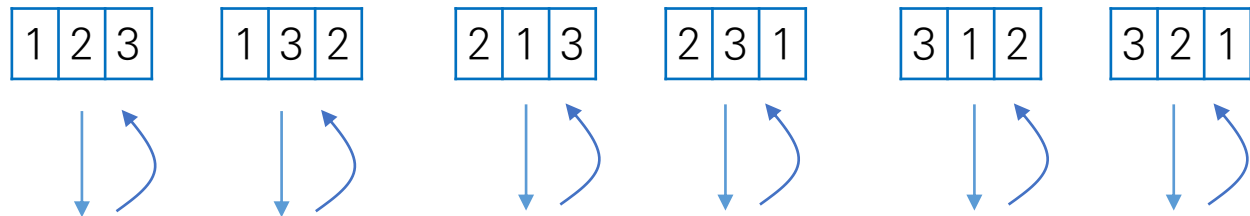
첫째 자리



둘째 자리



셋째 자리



완성

123

132

213

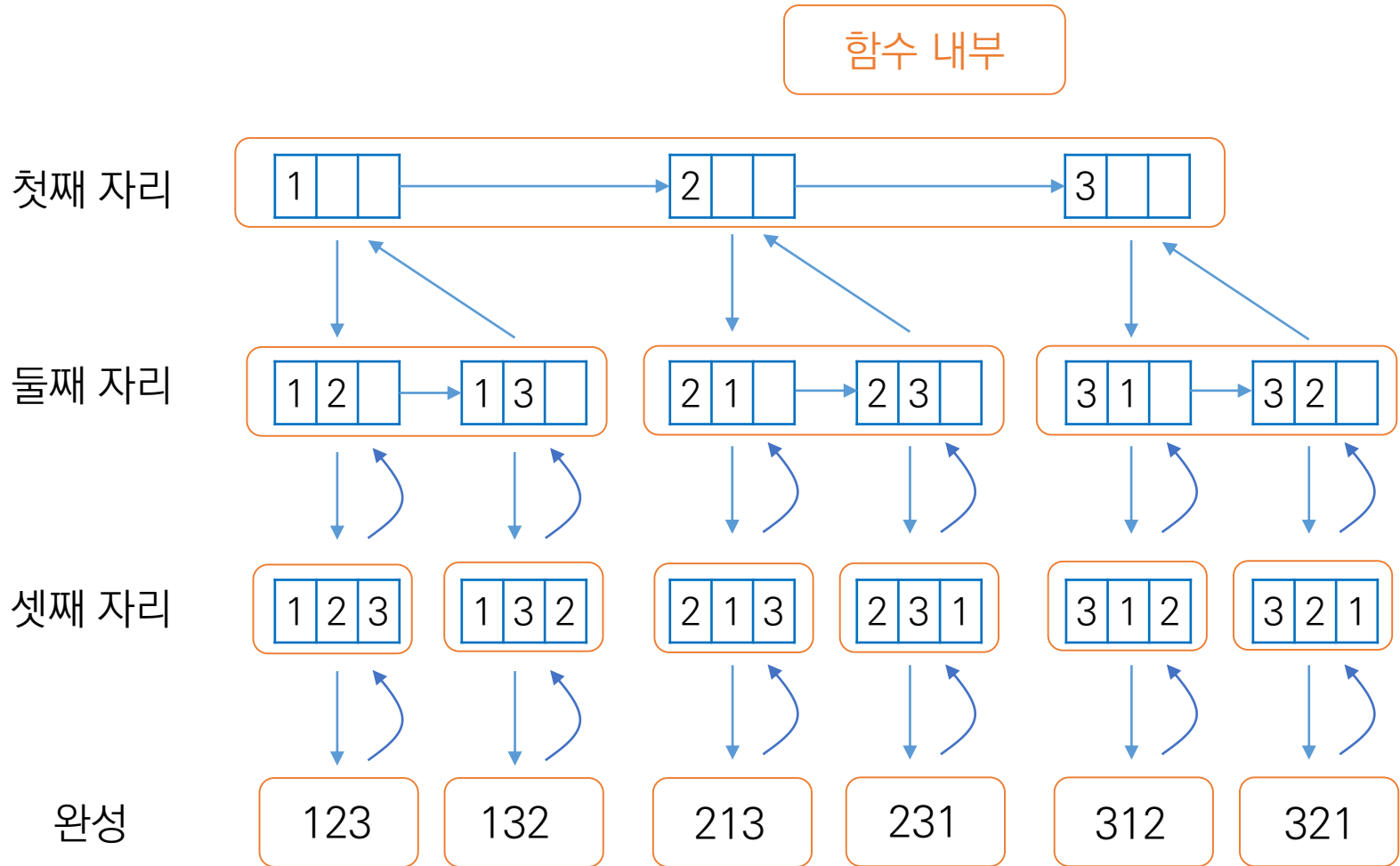
231

312

321



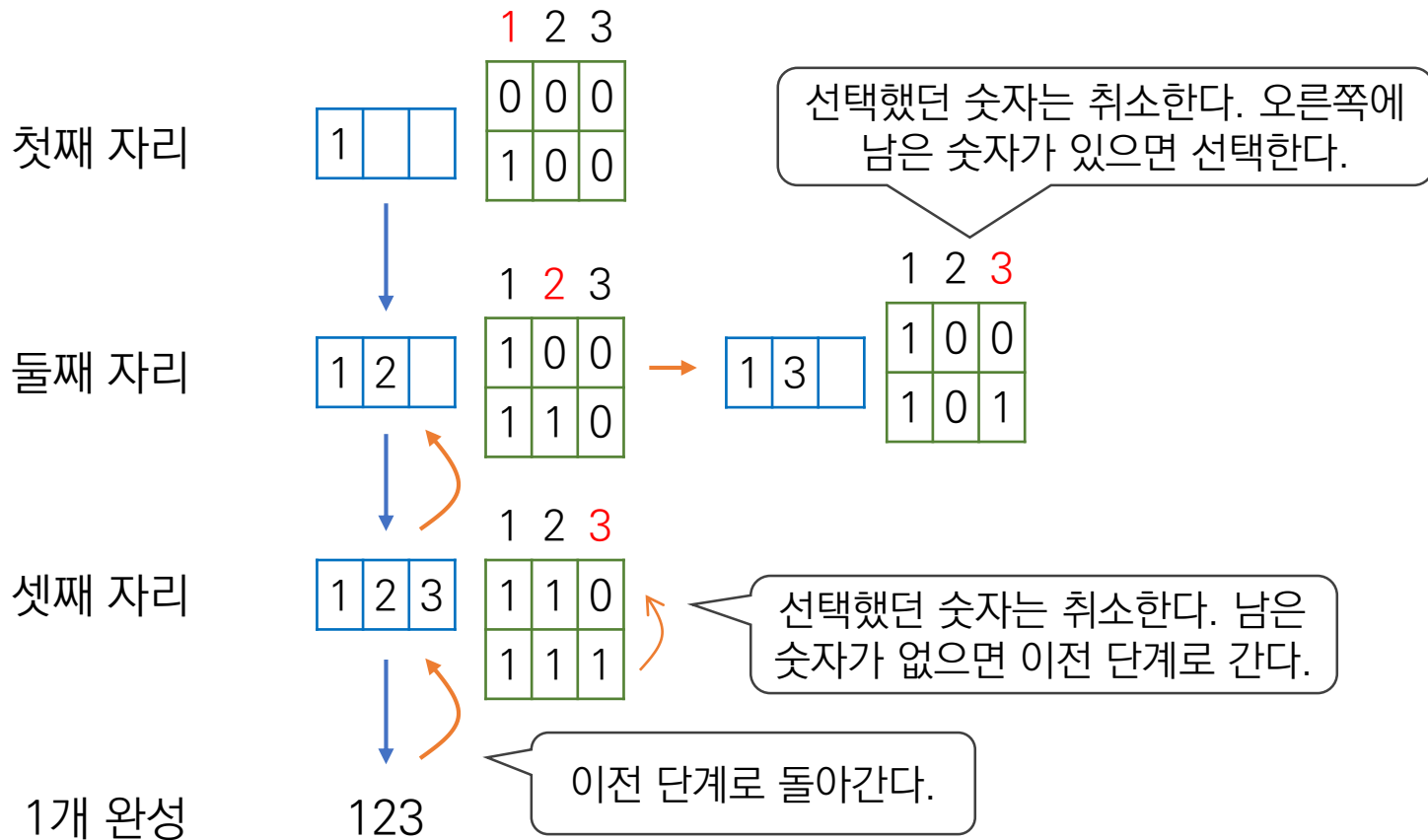
■ 1, 2, 3으로 3자리 수 만들기



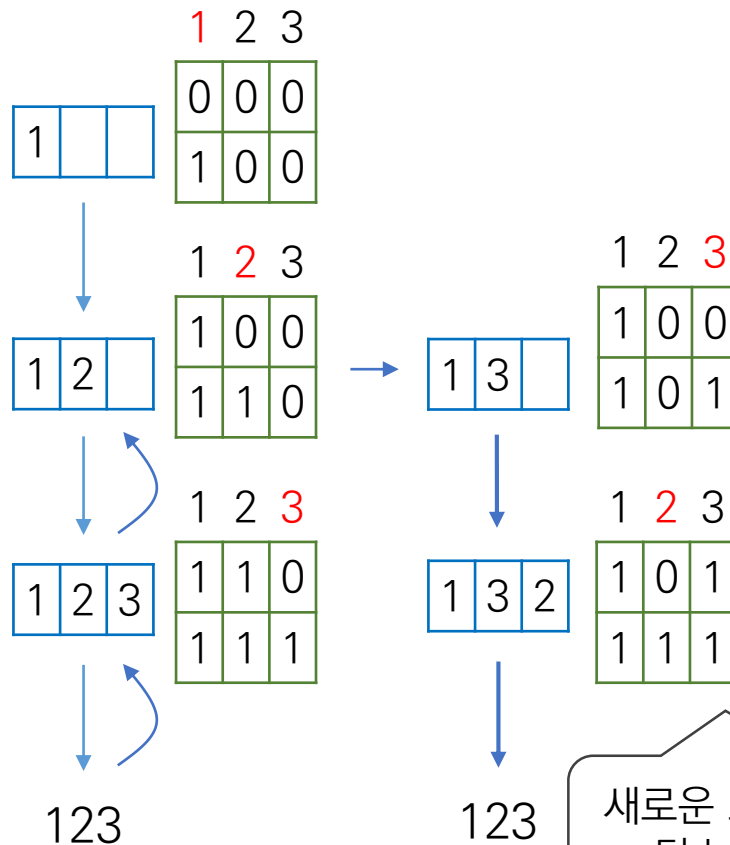
■ 1, 2, 3으로 3자리 수 만들기



■ 1, 2, 3으로 3자리 수 만들기 (계속)



■ 1, 2, 3으로 3자리 수 만들기 (계속)



n : 사용할 숫자 개수
k : 선택한 숫자를 넣을 자리
i : 사용하지 않은 숫자를 찾는 순서

```
for (int i = 0; i < n; i++)
{
    if (used[i] == 0)
    {
        used[i] = 1;
        p[k] = a[i];
        perm(n, k + 1);
        used[i] = 0;
    }
}
```

새로운 호출에서는 맨 왼쪽부터 남은 숫자를 찾는다.



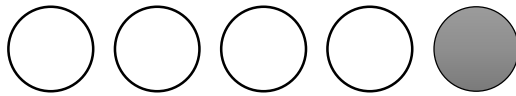
연습

- {1, 2, 3, 4}의 원소를 사용해 순열 만들기.



조합

- n 개에서 k 개를 고르는 경우의 수 : ${}_nC_k$
 - $\{1, 2, 3\}$ 에서 두 개의 숫자를 고르는 경우의 수 : ${}_3C_2$
 - $\{1, 2\}, \{1, 3\}, \{2, 3\}$
- 두 경우로 나눠 생각할 수 있다.



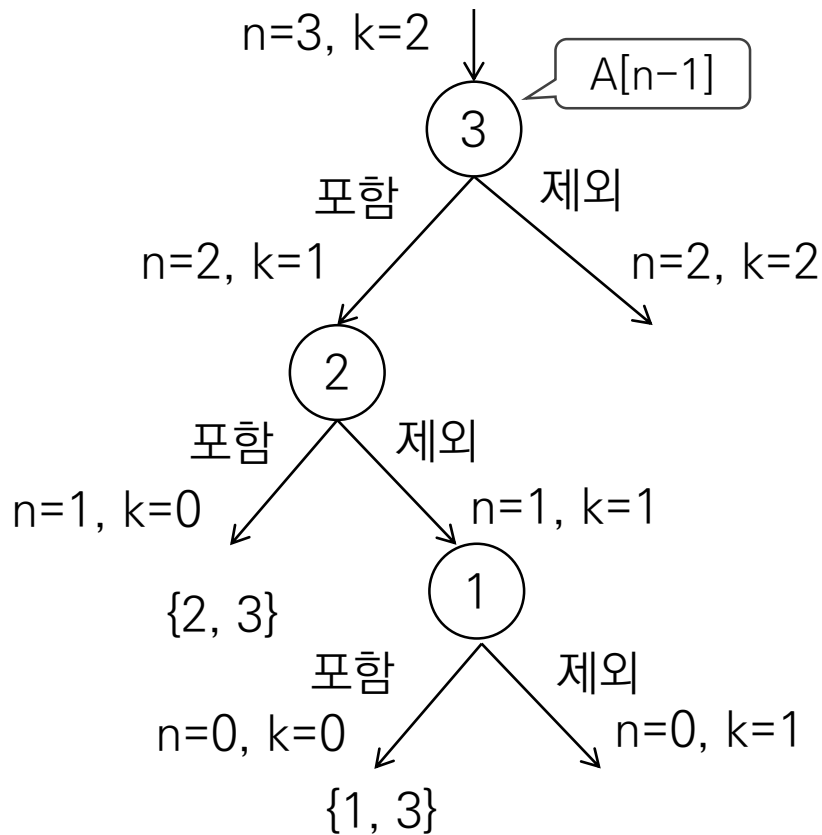
특정원소를 포함하는 경우와
포함하지 않는 경우로 나눠서
생각할 수 있음.

- 특정 원소를 선택하면 남은 숫자도 하나 줄고($n-1$), 골라야 하는 개수도 줄어듬($k-1$).
- 특정 원소를 제외하면 남은 숫자는 하나 줄고($n-1$), 골라야 하는 개수는 그대로(k).
- ${}_nC_k = {}_{n-1}C_{k-1} + {}_{n-1}C_k$

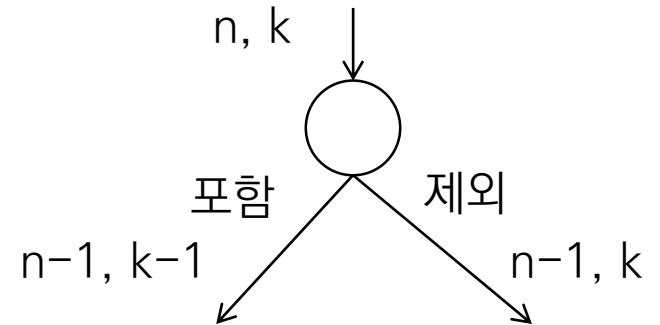


■ n 개에서 k 개를 고르는 경우의 수 : nCk

• $A[] = \{1, 2, 3\}$ 에서 두 개의 숫자를 고르는 경우의 수 : ${}_3C_2$



n : 선택가능한 개수
 k : 골라야 하는 개수



$n < k$ 은 조건에 맞지
않으므로 리턴



연습

- {1, 2, 3}에서 2개를 고르는 조합 만들기.

```
void ncr(int n, int k)
{
    if (k == 0)
    {
        for (int i = 0; i < 2; i++)
            printf("%d ", C[i]);
        printf("\n");
    }
    else if (n < k)
        return;
    else
    {
        C[k - 1] = A[n - 1];
        ncr(n - 1, k - 1);
        ncr(n - 1, k);
    }
}
```

