순열, 조합

Gaussian

순열(Permutation)

Permutation

- C++ 기준으로 순열은 next_permutation 함수를 사용하면 된다.
- 재귀함수로 작성하는 방법은 코드가 길어지므로 생략..
- next_permutation 사용

  ① 오름 차순으로 정렬한다.

  ② do{

  ...

  }while(next_permutation(A.begin(), A.end()))

  와 같이 사용하여 모든 순열을 방문한다.

- 모든 순열을 탐색할 때 O(N!)의 시간이 사용되므로 배열의 수가 많으면 사용하기 어렵다.
- N개의 원수 중 K개를 선택 하여 순열을 구현하려면 다음과 같다.

```
bool next_k_permutation(vector<int>&v, int k){
    sort(v.begin()+k, v.end(), greater<int>());
    return next_permutation(v.begin(), v.end());
}
```

조합(Combination)

Combination

- C++ 기준으로 조합은 next_permutation 함수 / 재귀로 쉽게 구현할 수 있다.

① 재귀를 이용하는 방법

pick : n개 중 toPick 개 만큼 선택하여 조합을 만드는 함수

```
void pick(int n, vector<int>& picked, int toPick){
    if(toPick==0) {
        printPicked(picked);
        return;
    }

    int smallest=picked.empty() ? 0 : picked.back()+1;
    for(int next=smallest; next<n; ++next){
        picked.push_back(next);
        pick(n,picked, toPick-1);
        picked.pop_back();
    }
}
```

조합(Combination)

Combination

- C++ 기준으로 조합은 next_permutation 함수 / 재귀로 쉽게 구현할 수 있다.

② next_permutation 이용

```
bool next_k_combination(vector<int>&v, int k){
    while(next_k_permutation(v,k)){
        if(is_sorted(v.begin(), v.begin()+k))
            return true;
    }
    return false;
}
```

- 차이를 최대로 : https://www.acmicpc.net/problem/10819
- 외판원 순회2 : https://www.acmicpc.net/problem/10971
- 로또 : https://www.acmicpc.net/problem/6603