

Baekjoon / N과 M (1) (15649번)

문제

자연수 N 과 M 이 주어졌을 때, 아래 조건을 만족하는 길이가 M 인 수열을 모두 구하는 프로그램을 작성하시오.
1부터 N 까지 자연수 중에서 중복 없이 M 개를 고른 수열

입력

첫째 줄에 자연수 N 과 M 이 주어진다. ($1 \leq M \leq N \leq 8$)

출력

한 줄에 하나씩 문제의 조건을 만족하는 수열을 출력한다. 중복되는 수열을 여러 번 출력하면 안되며, 각 수열은 공백으로 구분해서 출력해야 한다.

수열은 사전 순으로 증가하는 순서로 출력해야 한다.

예제 입력 1

3 1

예제 출력 1

1
2
3

예제 입력 2

4 2

예제 출력 2

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

예제 입력 3

4 4

예제 출력 3

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

문제 접근 방법

1. 1부터 N까지의 수를 탐색할 수 있도록 반복문을 설정한다
→ 수열은 항상 1부터 시작하므로, `for (int i = 1; i <= N; ++i)` 형태로 반복을 돈다.
2. 백트래킹(재귀 함수)를 사용하여 수열을 하나씩 만든다
→ 현재 선택된 수열의 길이가 M이 될 때까지 숫자를 하나씩 선택하고, 조건이 맞으면 다음 단계로 넘어간다.
3. 중복된 수를 선택하지 않기 위해 사용한 수를 체크한다
→ `vector<bool> used`를 만들어 이미 선택한 수는 다시 선택하지 않도록 한다.
4. 수열의 길이가 M이 되면 출력한다
→ 수열이 완성됐을 경우 (`k == M`) 지금까지 선택된 수들을 출력한다.
5. 백트래킹을 통해 이전 상태로 돌아가 다른 수를 선택할 수 있게 한다
→ 재귀가 끝나면 마지막에 선택한 수를 제거하고(`pop_back()`), 해당 수를 미사용 처리하여 다른 조합을 만들 수 있도록 한다.

코드

```
// 15649번 / N과 M (1) / baekjoon
24ms, 1h

#include <iostream>
#include <vector>

using namespace std;

int N, M;
vector<int> result;
vector<bool> used;

void make(int k) {
    if (k == M) {
        for (int num : result)
            cout << num << " ";
        cout << "\n";
        return;
    }

    for (int i = 1; i <= N; ++i) {
        if (!used[i]) {
            used[i] = true;
            result.push_back(i);
            make(k + 1);
            result.pop_back();
            used[i] = false;
        }
    }
}

int main() {
    cin >> N >> M;
    used.resize(N + 1, false);
    make(0);
    return 0;
}
```