

N & M

< N과 M > 문제의 특성과 틀

N과 M (1) 문제와 -

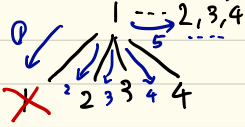
• 1부터 N까지 자연수 중에서 중복 없이

M개를 고를 수열이다.

↳ 부호는 문제마다 달라지는 부분!

↳ input data가 어떤 역할인지.

N과 M은 (input 4, 2 가정)



중복성

화살표 를 보았을때 깊이우선타입식(BFS)

이며, X 중복 제거가 필요하다!

```
void foo (int d) {
```

```
    if (d == M) { // 종료조건
```

```
        원소 출력
```

```
        return
```

```
    }
```

```
    else { // to do
```

```
        for (int i = 0; i < N; i++) { // 재귀호출
```

```
            if (!check[i]) { // 중복 X.
```

```
                check[i] = true;
```

```
                vector.push_back(i);
```

```
                foo(d+1); // 재귀호출 (깊이+1)
```

```
                check[i] = false;
```

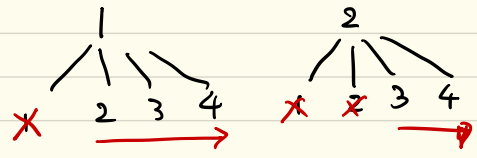
```
                vector.pop_back();
```

기본적인 틀은 앞에 코드와 같다.

이후 문제들은 조건에 따라 다뤄 보자

< N과 M (2) > - 앞편에서 '조건'은 '오류'라는 개념 추가!

트리구조를 짜 보자. (input 4, 2 가정)



부모노드가 자식노드보다 무조건 작다!

⇒ for문을 부모노드 값 다음부터 시작!

⇒ 재귀가 없어도 부모노드 index 필요

⇒ 함수 파라미터 추가!

```
void foo (int d, int now)
```

```
{ (선택)
```

```
else {
```

```
    for (int i = now; i < N; i++)
```

```
        if (!check[i]) {
```

```
            ;
```

```
            foo(d+1, now)
```

```
            ;
```

```
        }
```

<N과 M (3)>

같은 수를 여러번 골라도 된다!
N과 M(1) 코드에서 중복 체크 부분을
제거해버리기!

<N과 M (4)>

중복은 가능하나 부모노드부터 탐색!
↳ N과 M (3) ↳ N과 M (2)

중복 체크 없애고, N과 M (2)처럼 now
파라미터 추가!

<N과 M (5)>

예제 1... N이 아닌, N개의 수가
주어진다는!

그럼 같은 수들의 : 같은 원소가 아닌
주어진 값이 있는 vector의 index가 되면
된다. ↳ (생각해줘야함)

<N과 M (1) 코드에서...>

사용자 입력된 수를 vector에 넣고,
v.push_back(node[i]);로 변형해
주면 끝이다.

<N과 M (6)>

N과 M(5) + N과 M(2)

↳ 사용자 input ↳ 부모노드 부터 탐색

<N과 M (7)>

N과 M(5) + N과 M(3)

<N과 M (8)>

N과 M(5) + N과 M(4)

<N과 M (9)> N과 M(5) + Now

점진적 배열이 꼭 들어간다..

왜 그렇냐면, 주어진 수가 중복이 되기 때문

→ 중복 체크 unique 쓰면 X
(test case 99 수리 X)

→ 설정된 수열을 기록해서 기록한 것이 없으면 출력!

vector <vector<int>> record; // 수열 저장

for(int i = 0; i < binary_search(record.begin(), record.end(), v); i++)

if (i < binary_search(record.begin(), record.end(), v))
for (int j = 0; j < record[i].size(); j++)
cout << ...

안되면 false
O(log N)
find 함수도 O(N)

record.push_back(v);

}

이런식으로 해줘야한다.

< N과 M (10) >

$$N과 M(9) + N과 M(5) + N과 M(2)$$

< N과 M (11) >

$$N과 M(9) + N과 M(6) + N과 M(3)$$

< N과 M (12) >

$$N과 M(9) + N과 M(5) + N과 M(4)$$

이 N과 M들 안에서 다스 문제는 다
알았다고 한다!

77
—
6

—