재귀, 백트래킹, 분할 정복

수학적 귀납법

- $P(1)$이 참이고, $P(n-1) \rightarrow P(n)$이 참이면, $P(n)$은 모든 자연수 n에 대해서 참이다.

- 직관과는 거리가 있지만, 수학적으로 증명하는 것에 익숙해져야 함

수학적 귀납법과 재귀

- $P(1)$이 참이고, $P(n-1) \rightarrow P(n)$이 참이면, $P(n)$은 모든 자연수 n에 대해서 참이다.

- 1부터 x까지의 합을 구하는 함수가 정말 우리가 원하는 값을 리턴하는지 증명해 보자.

```
int sum(int x){
    if (x <= 0) return 0;
    return x + sum(x-1);
}
```

수학적 귀납법과 재귀

- $P(1)$이 참이고, $P(n-1) \rightarrow P(n)$이 참이면, $P(n)$은 모든 자연수 n에 대해서 참이다.

- 1부터 x까지의 합을 구하는 함수가 정말 우리가 원하는 값을 리턴하는지 증명해 보자.

```
int sum(int x){
    if (x <= 0) return 0;
    return x + sum(x-1);
}
```

$$
\begin{aligned}
&\texttt{sum}(4) \\
&\rightarrow 4 + \texttt{sum}(3) \\
&\rightarrow 4 + (3 + \texttt{sum}(2)) \\
&\rightarrow 4 + (3 + (2 + \texttt{sum}(1))) \\
&\rightarrow 4 + (3 + (2 + (1 + \texttt{sum}(0)))) \\
&\rightarrow 4 + (3 + (2 + (1 + 0))) \\
&\rightarrow 10
\end{aligned}
$$

수학적 귀납법과 재귀

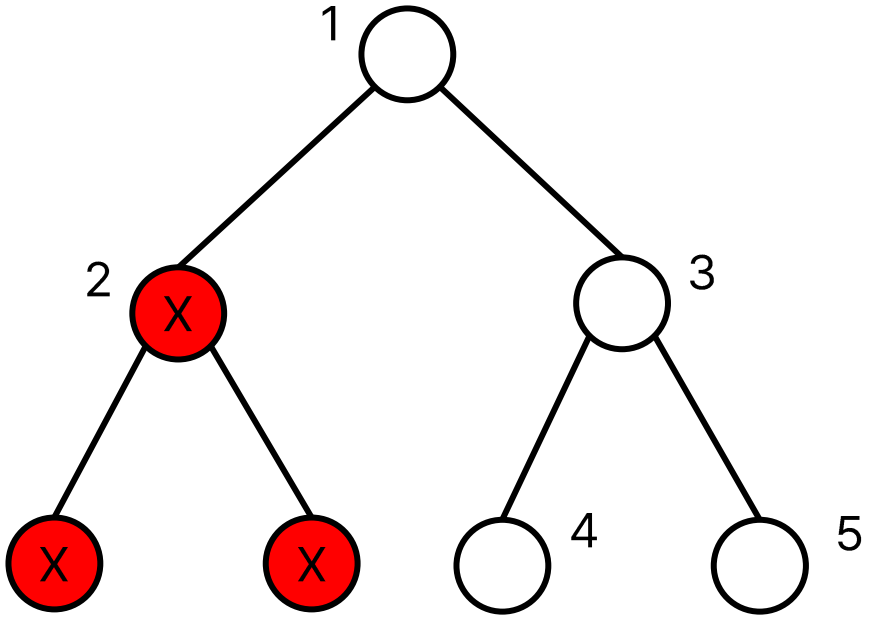
- Base: $x = 0$일 때, 합은 0이다. (참)
- Step: $x = k - 1$일 때, 함수가 1부터 $k - 1$까지의 합을 리턴한다면,
  그 값에 k를 더한 값은 1부터 k까지의 합과 같다.

- $P(1)$이 참이고, $P(n-1) \rightarrow P(n)$이 참이면, $P(n)$은 모든 자연수 n에 대해서 참이다.
- 만약 위 식에서 $P(n-1)$이 거짓이라면, $P(n)$은 참이 되기 때문에 생각할 필요가 없다.
- 따라서 $P(n-1)$이 참이라고 가정해도 된다.

```
int sum(int x){
    if (x <= 0) return 0;
    return x + sum(x-1);
}
```

백트래킹

- 해를 찾는 도중 해가 아니라면 되돌아가서 다시 해를 찾아가는 기법
- 완전탐색을 하는 과정에서 **조건에 맞지 않는** 부분을 **가지치기**함

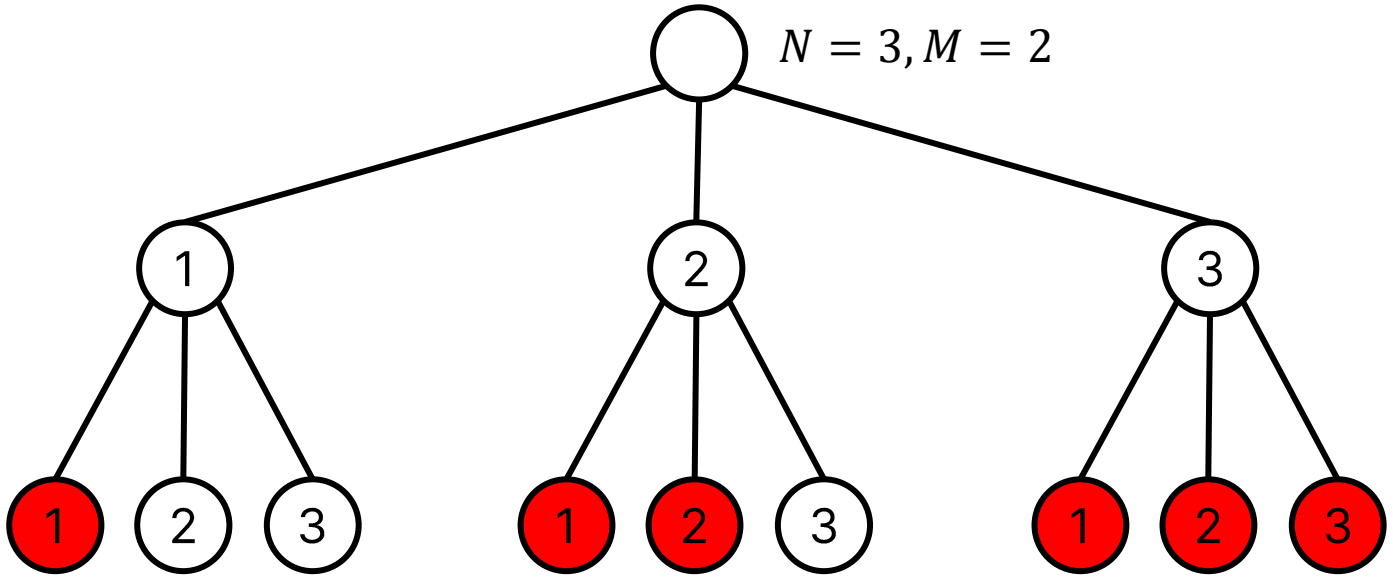


N과 M (2) BOJ 15650

- 자연수 N과 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 구해보자.
  - 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열
  - 고른 수열은 오름차순이어야 한다.
- 수열은 사전 순으로 증가하는 순서로 출력한다.

N과 M (2) BOJ 15650

- 자연수 N과 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 구해보자.
  - 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열
  - 고른 수열은 오름차순이어야 한다.



N과 M (2) BOJ 15650

```
void dfs(int x, int cnt) {
    if (cnt == M) {
        for (int i = 0; i < M; i++) {
            cout << A[i] << " ";
        }
        cout << "\n";
        return;
    }
    for (int i = x + 1; i <= N; i++) {
        A[cnt] = i;
        dfs(i, cnt + 1);
    }
}
```

해를 찾은 경우

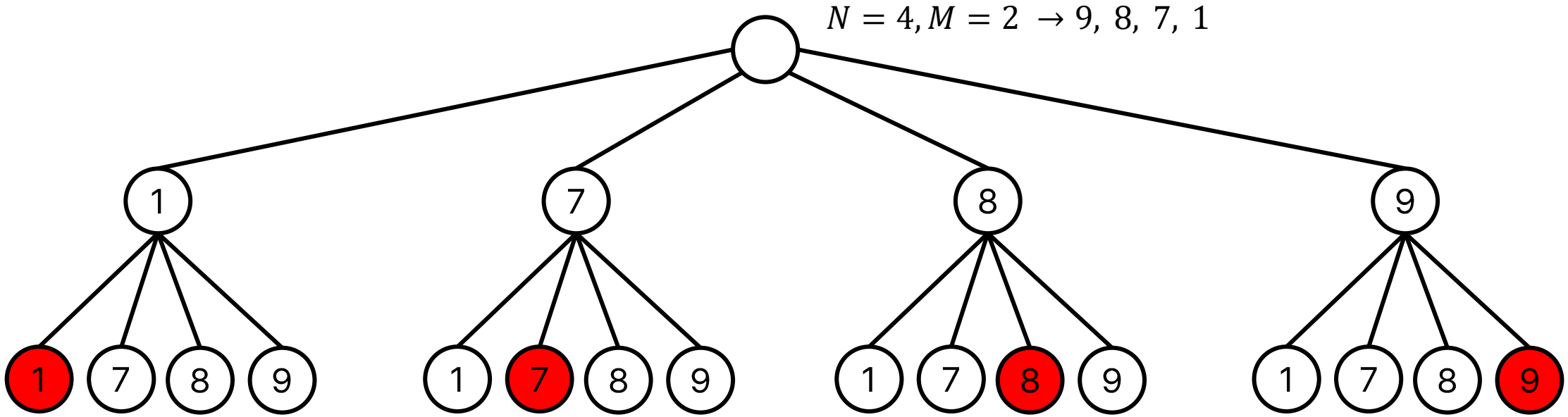
가지치기

N과 M (5) BOJ 15654

- N개의 자연수와 자연수 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구해보자. N개의 자연수는 모두 다른 수이다.
  - N개의 자연수 중에서 M개를 고른 수열
- 수열은 사전 순으로 증가하는 순서로 출력한다.

N과 M (5) BOJ 15654

- N개의 자연수와 자연수 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구해보자. N개의 자연수는 모두 다른 수이다.
  - N개의 자연수 중에서 M개를 고른 수열



N과 M (5) BOJ 15654

```
void dfs(int cnt) {
    if (cnt == M) {
        for (int i = 0; i < M; i++)
            cout << A[i] << " ";
        }
        cout << "\n";
        return;
    }
    for (int i = 0; i < N; i++) {
        if (!vst[i]) {
            vst[i] = 1; A[cnt] = v[i];
            dfs(cnt + 1);
            vst[i] = 0;
        }
    }
}
```

해를 찾은 경우

주어진 자연수들이 정렬되어 있음

가지치기

중복 체크

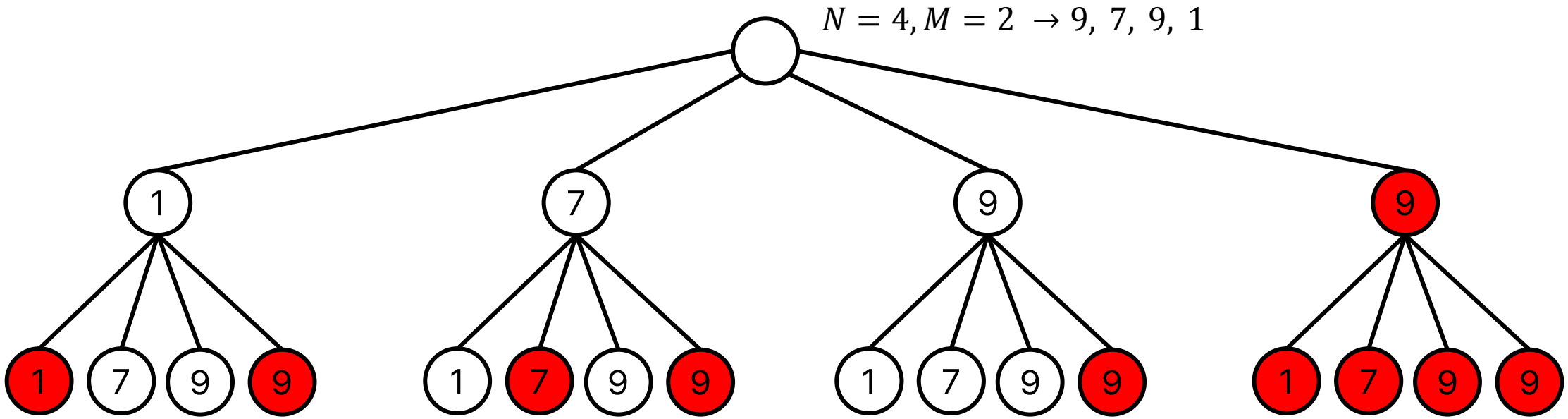
중복 체크 해제

N과 M (9) BOJ 15663

- N개의 자연수와 자연수 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구해보자.
  - N개의 자연수 중에서 M개를 고른 수열
- 수열은 사전 순으로 증가하는 순서로 출력한다.

N과 M (9) BOJ 15663

- N개의 자연수와 자연수 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구해보자.
  - N개의 자연수 중에서 M개를 고른 수열



N과 M (9) BOJ 15663

```
void dfs(int cnt) {
    if (cnt == M) {
        for (int i = 0; i < M; i++)
            cout << A[i] << " ";
        }
        cout << "\n";
        return;
    }
    int prev = -1;
    for (int i = 0; i < N; i++) {
        if (!vst[i] && prev != v[i]) {
            vst[i] = 1; A[cnt] = v[i];
            dfs(cnt + 1);
            vst[i] = 0; prev = v[i];
        }
    }
}
```

해를 찾은 경우

주어진 자연수들이 정렬되어 있음

가지치기

중복 체크

중복 체크 해제

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자
- N개를 놓기 위해서 어떻게 놓아야 할까?

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자
- N개를 놓기 위해서 어떻게 놓아야 할까?
  - 1줄에 1개는 반드시 있어야 한다.

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자

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

→

조건에 맞지 않음

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

→

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

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자

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

→

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

→

조건에 맞지 않음

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

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자

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

→

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

→

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

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \le N \le 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자

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

→

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

모두 배치하였으므로,
답을 증가시키고 1, 3부터...

N-Queen BOJ 9963

- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자
- 놓을 수 있는지 없는지는 어떻게 검사할까?

N-Queen BOJ 9963

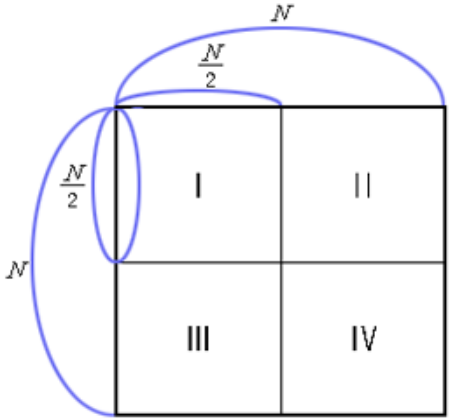
- 크기가 $N \times N$인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제
- $N(1 \leq N \leq 15)$이 주어졌을 때, 퀸을 놓는 방법의 수를 구해보자
- 놓을 수 있는지 없는지는 어떻게 검사할까?
  - 행 순서대로 하나씩 배치를 하였기에, 열 또는 대각선 위치인지만 체크!
    - 열 : 현재 퀸의 열과 다른 퀸들의 열이 같은가?
    - 대각선 : (다른 퀸의 행 - 현재 퀸의 행)과 |다른 퀸의 열 - 현재 퀸의 열|값이 같은가?
  - 이렇게 검사를 한다면 1차원 배열로도 해결이 가능하다.

분할 정복

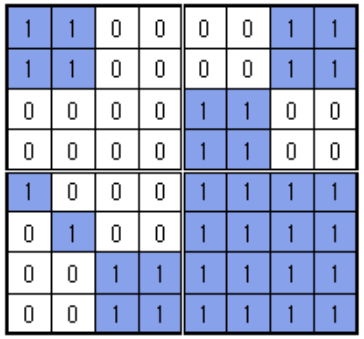
- 주어진 문제를 여러 개의 작은 문제로 **분할**
- 작은 문제들의 답을 이용해 주어진 문제의 답을 구함 (**정복**)

색종이 만들기 BOJ 2630

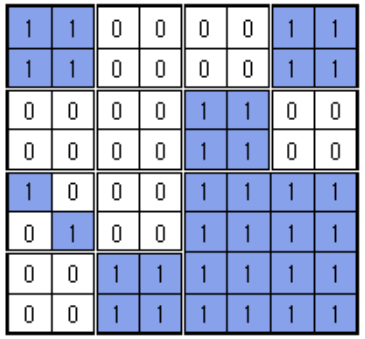
- 가장 작은 문제는 무엇일까?
- 큰 문제를 어떻게 작은 문제로 쪼갤까?



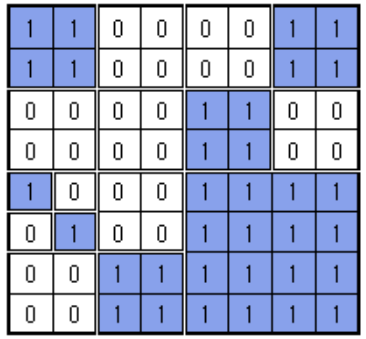
<그림 2>



<그림 3> 처음 나눈 후의 상태



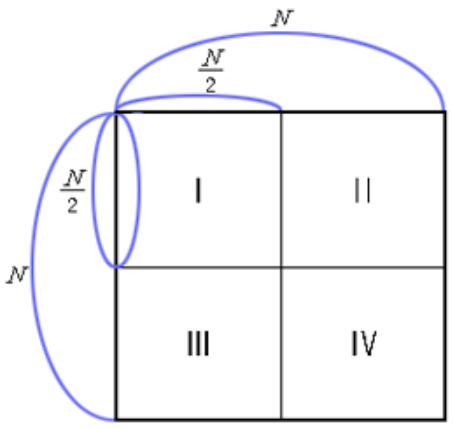
<그림 4> 두 번째 나눈 후의 상태



<그림 5> 최종적으로 나누어진 색종이들

색종이 만들기 BOJ 2630

- 가장 작은 문제는 무엇일까?
  - 1×1크기의 종이이다.
- 큰 문제를 어떻게 작은 문제로 쪼갤까?
  - 그림과 같이 I, II, III, IV의 종이로 쪼갠다.
  - 종이는 시작 위치와 크기로 정의할 수 있다.



<그림 2>

1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 3> 처음 나눈 후의 상태

1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 4> 두 번째 나눈 후의 상태

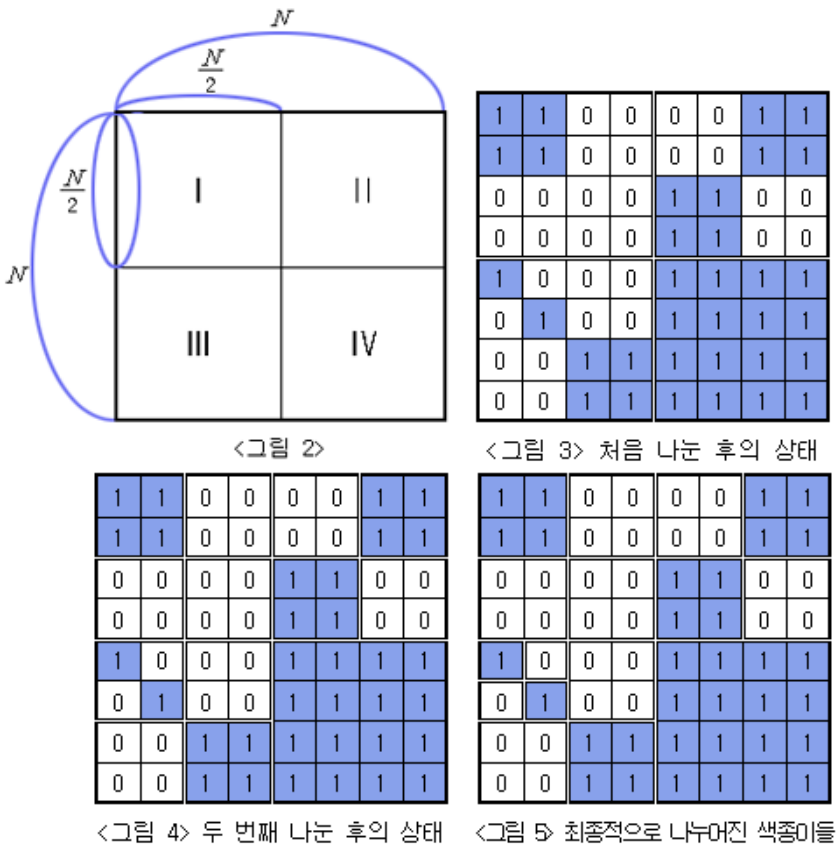
1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 5> 최종적으로 나누어진 색종이들

색종이 만들기 BOJ 2630

- 가장 작은 문제는 무엇일까?
  - 1×1크기의 종이이다.
- 큰 문제를 어떻게 작은 문제로 쪼갤까?
  - 그림과 같이 I, II, III, IV의 종이로 쪼갠다.
  - 종이는 시작 위치와 크기로 정의할 수 있다.

```
solve(x, y, size / 2);
solve(x + size / 2, y, size / 2);
solve(x, y + size / 2, size / 2);
solve(x + size / 2, y + size / 2, size / 2);
```



<그림 2>

<그림 3> 처음 나눈 후의 상태

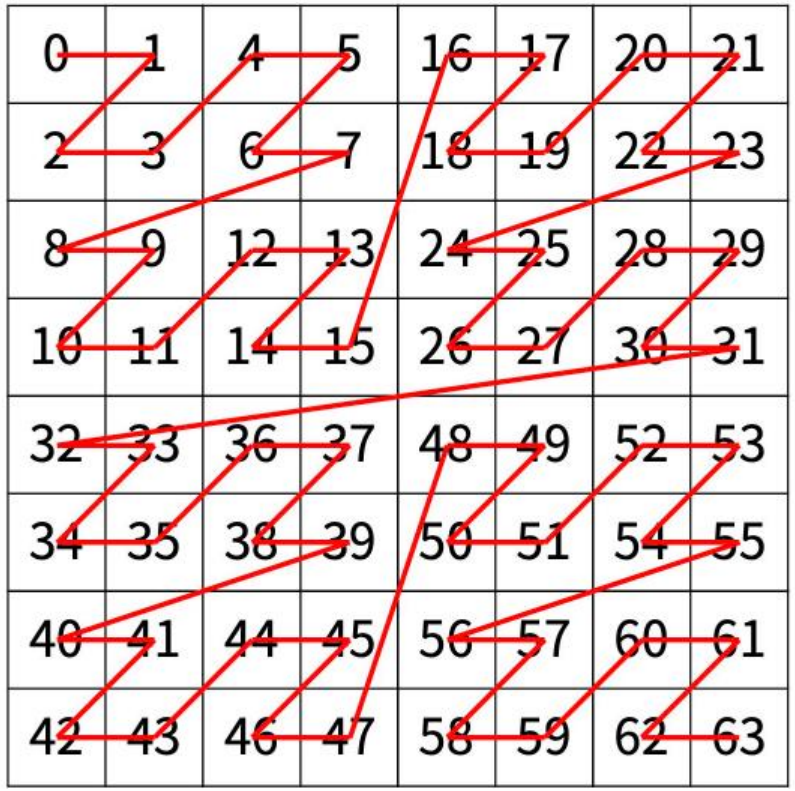
<그림 4> 두 번째 나눈 후의 상태

<그림 5> 최종적으로 나누어진 색종이들

Z BOJ 1074

- r행 c열을 몇 번째로 방문하는지 출력하는 문제이다.
- 가장 작은 문제는 무엇일까?
- 큰 문제를 어떻게 작은 문제로 쪼갤까?

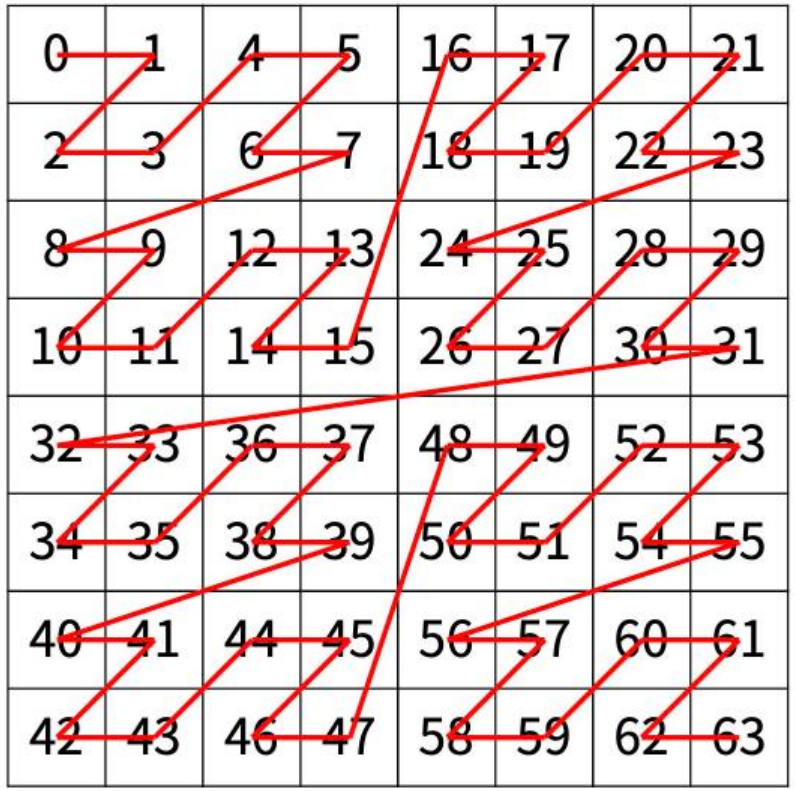
$N = 3$



Z BOJ 1074

- r행 c열을 몇 번째로 방문하는지 출력하는 문제이다.
- 가장 작은 문제는 무엇일까?
  - 1×1크기의 배열이다.
- 큰 문제를 어떻게 작은 문제로 쪼갤까?
  - 크기가 $2^{N-1} \times 2^{N-1}$인 배열로 4등분 한다.

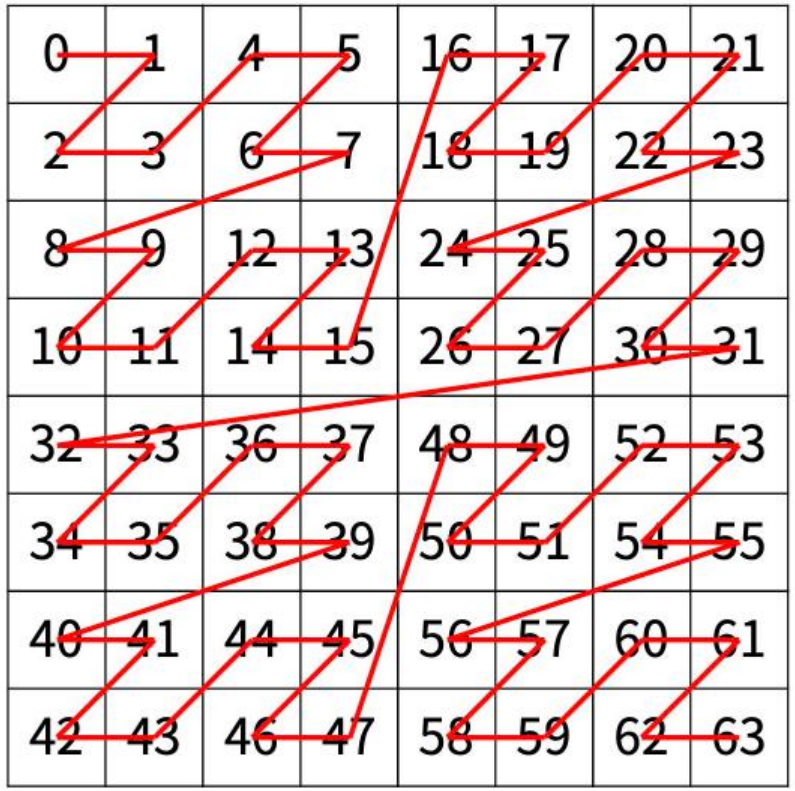
$N = 3$



Z BOJ 1074

- r행 c열을 몇 번째로 방문하는지 출력하는 문제이다.
- 몇 번째로 방문하는지는 어떻게 구할까?

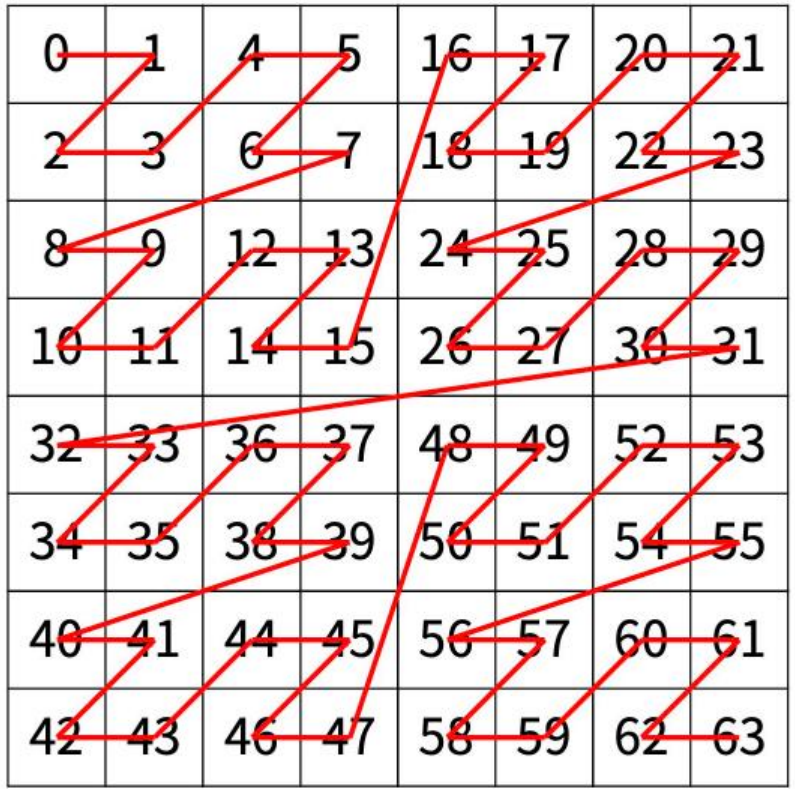
$N = 3$



Z BOJ 1074

- r행 c열을 몇 번째로 방문하는지 출력하는 문제이다.
- 몇 번째로 방문하는지는 어떻게 구할까?
  - 문제를 쪼갤 때, 방문하려는 곳이 쪼개려는 곳에 포함된다면?
  - 포함되지 않는다면?

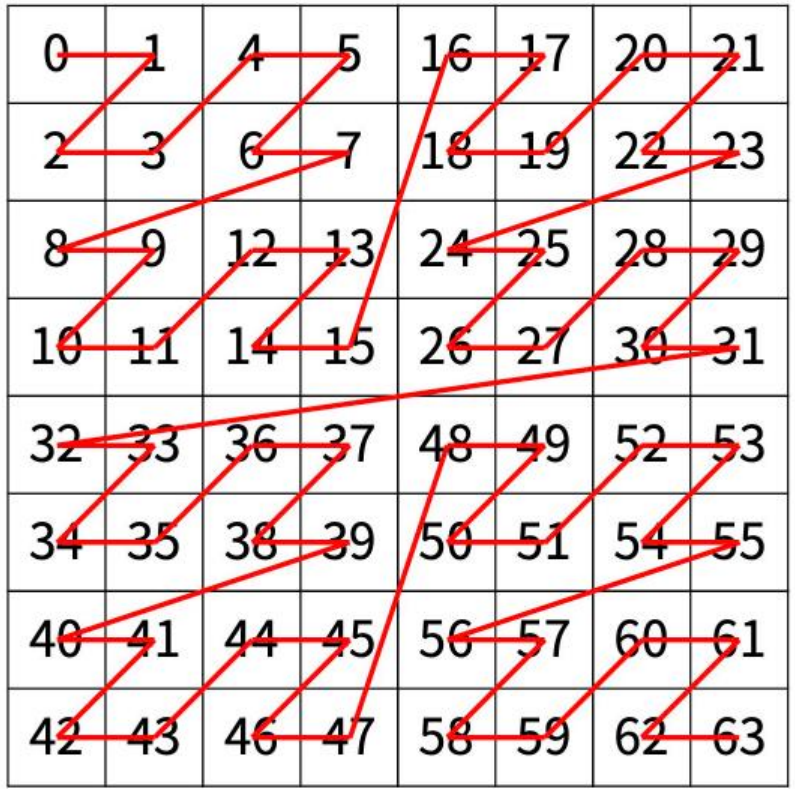
$N = 3$



Z BOJ 1074

- r행 c열을 몇 번째로 방문하는지 출력하는 문제이다.
- 몇 번째로 방문하는지는 어떻게 구할까?
- 문제를 쪼갤 때, 방문하려는 곳이 쪼개려는 곳에 포함된다면?
  - 좌상단, 우상단, 좌하단, 우하단 순으로 쪼개면 된다.
- 포함되지 않는다면?
  - 해당 배열의 크기만큼 답에 더해준다.

$N = 3$



쉬운 곱셈

- $A^k \ mod \ M$을 구하자.

곱셈 BOJ 1629

- $A^k \ mod \ M$을 구하자. $(A, M, k \leq 2\ 147\ 483\ 647)$

곱셈 BOJ 1629

- $A^k \ mod \ M$을 구하자. $(A, M, k \leq 2\ 147\ 483\ 647)$

- Naïve approach

$$A \times A \times A \times A \times \cdots \times A \ mod \ M$$

- Overflow

$$\left(\left(\left((A \times A) \ mod \ M\right) \times A\right) mod \ M\right) \times \cdots$$

- 시간복잡도는 $O(N)$

결과
시간 초과

곱셈 BOJ 1629

- $A^k \ mod \ M$을 구하자. $(A, M, k \leq 2\ 147\ 483\ 647)$
- $A^k = (A^2)^{\frac{k}{2}}$ 임을 활용할 수 있다
- 2^8을 단순하게 계산한다면 8번의 연산이 필요하다
- $2^8 = (2^2)^4 = ((2^2)^2)^2$
- 제곱 한 번이 계산이므로 총 3번의 계산으로 2^8을 계산할 수 있다.

곱셈 BOJ 1629

- $A^k \ mod \ M$을 구하자. $(A, M, k \leq 2\ 147\ 483\ 647)$

- $f(n, k)$가 n^k를 계산해준다고 하자.

- Base: $k = 0$인 경우, 1을 리턴한다. (참)
- Step: $k \geq 1$인 짝수의 경우, $f(n, \ \frac{k}{2}) \times f\left(n, \ \frac{k}{2}\right)$를 기계적으로 계산하면 된다.
  $k \geq 1$인 홀수의 경우, $n \times f(n, \ \ k-1)$를 기계적으로 계산하면 된다.

- 나머지 연산은 분배법칙을 활용해 오버플로우나지 않도록 해주자.

곱셈 BOJ 1629

- $A^k \ mod \ M$을 구하자. $(A, M, k \leq 2\ 147\ 483\ 647)$

```
long long f(long long a, long long b){
    if (b == 0) return 1;
    ll val = f(a, b / 2);
    if (b % 2) return a * val * val;
    else return val * val;
}
```

→ 충분히 작은 문제는 직접 해결

→ 큰 문제는 작은 문제로 쪼갬

→ 작은 문제의 정답으로 큰 문제의 답을 구함

- 시간복잡도 $T(k) = T\left(\frac{k}{2}\right) + O(1) = O(logk)$

연습 문제

15650 : N과 M (2)
15654 : N과 M (5)
15663 : N과 M (9)
9663 : N-Queen
2630 : 색종이 만들기
1074 : Z
1629 : 곱셈