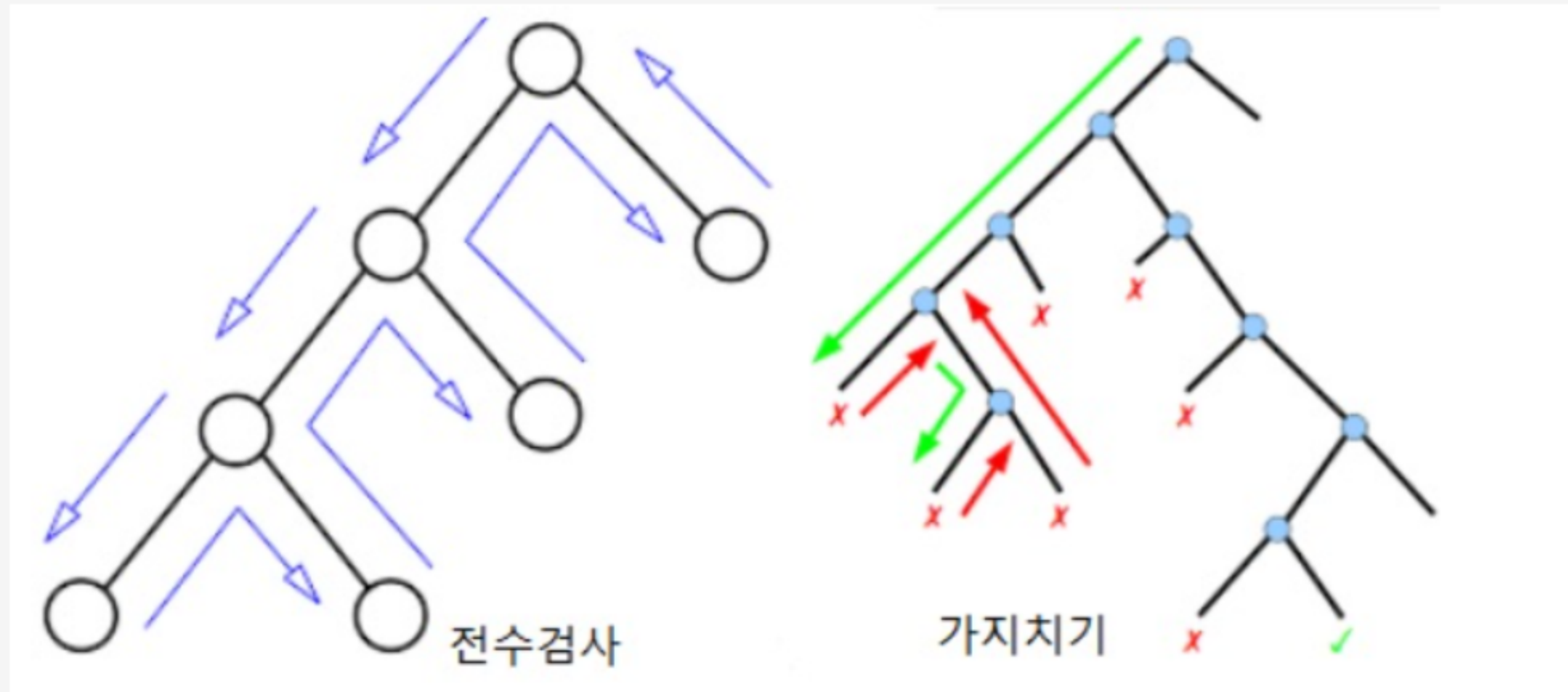


02. 완전탐색(백트래킹)

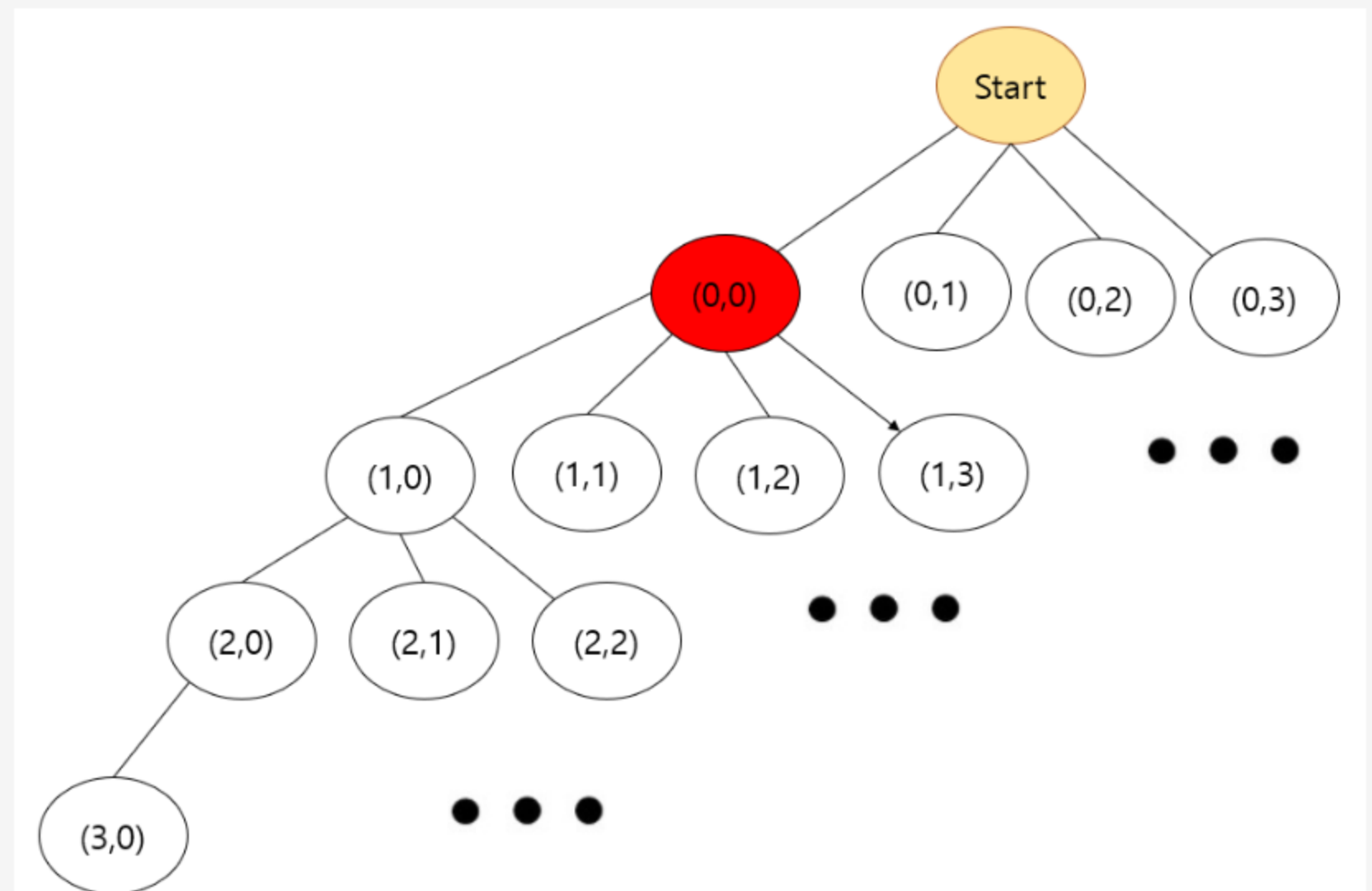
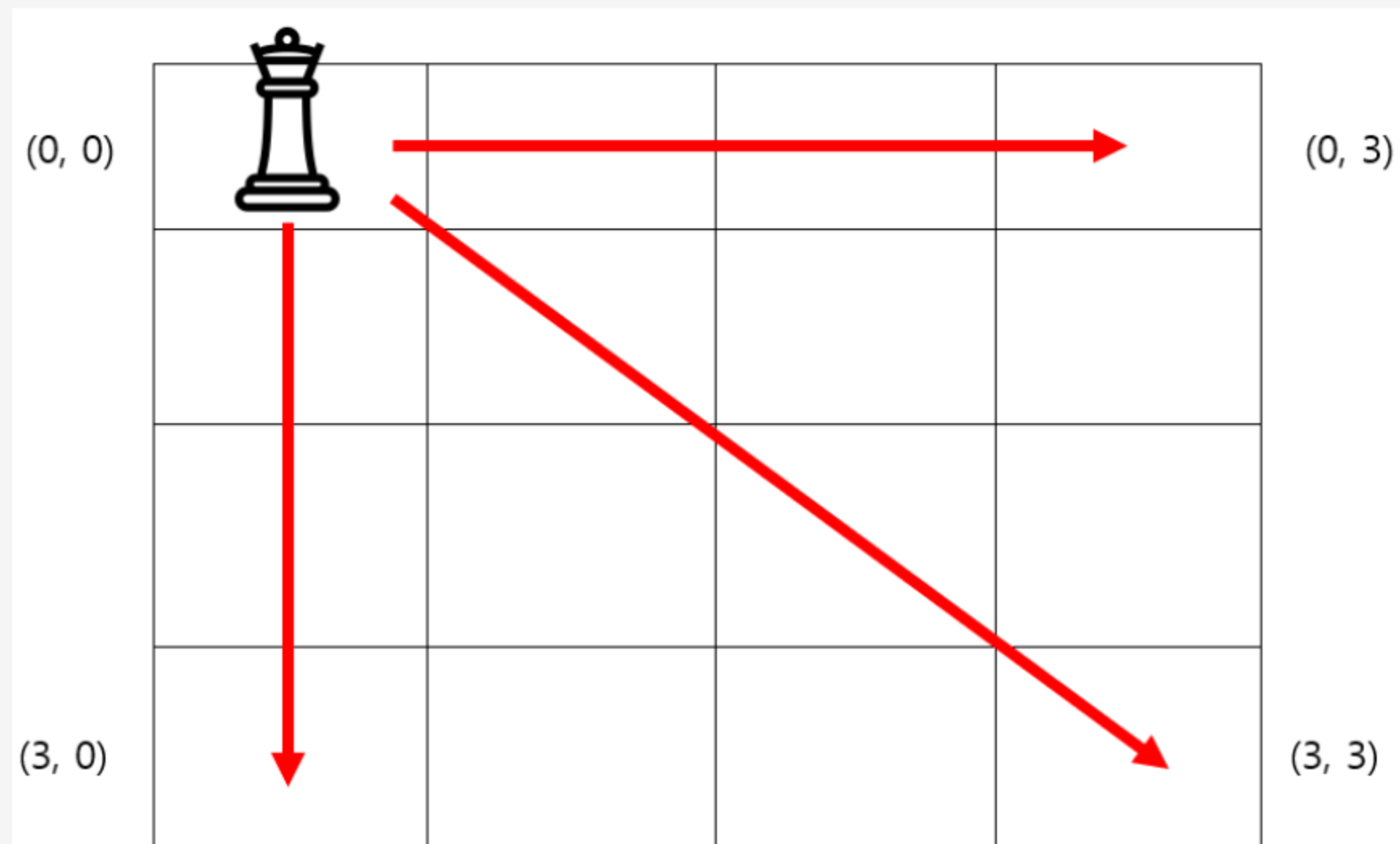
백트래킹이란?

완전 탐색을 하는 도중, 현재 탐색이 무의미한 경우 되돌아가는 알고리즘



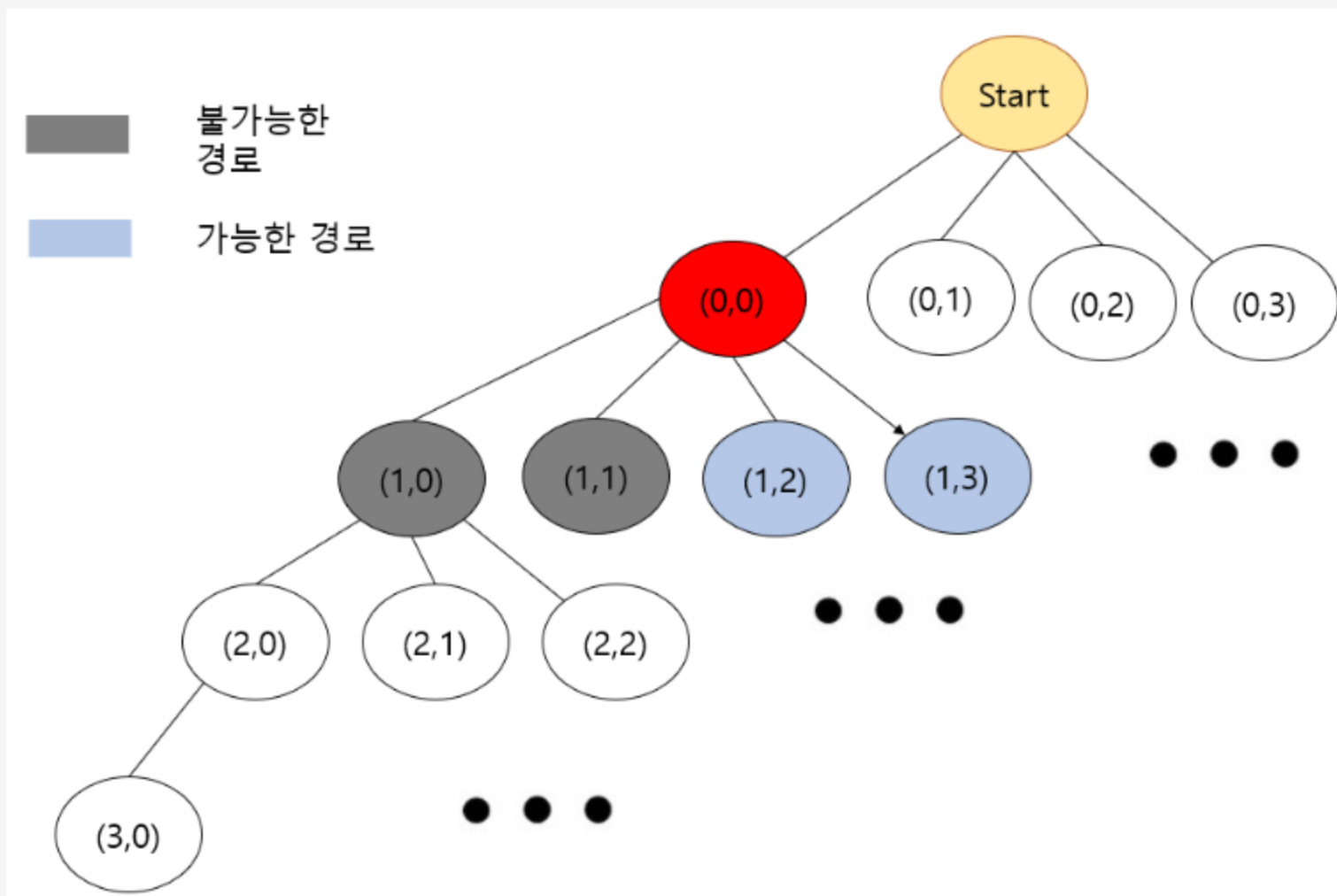
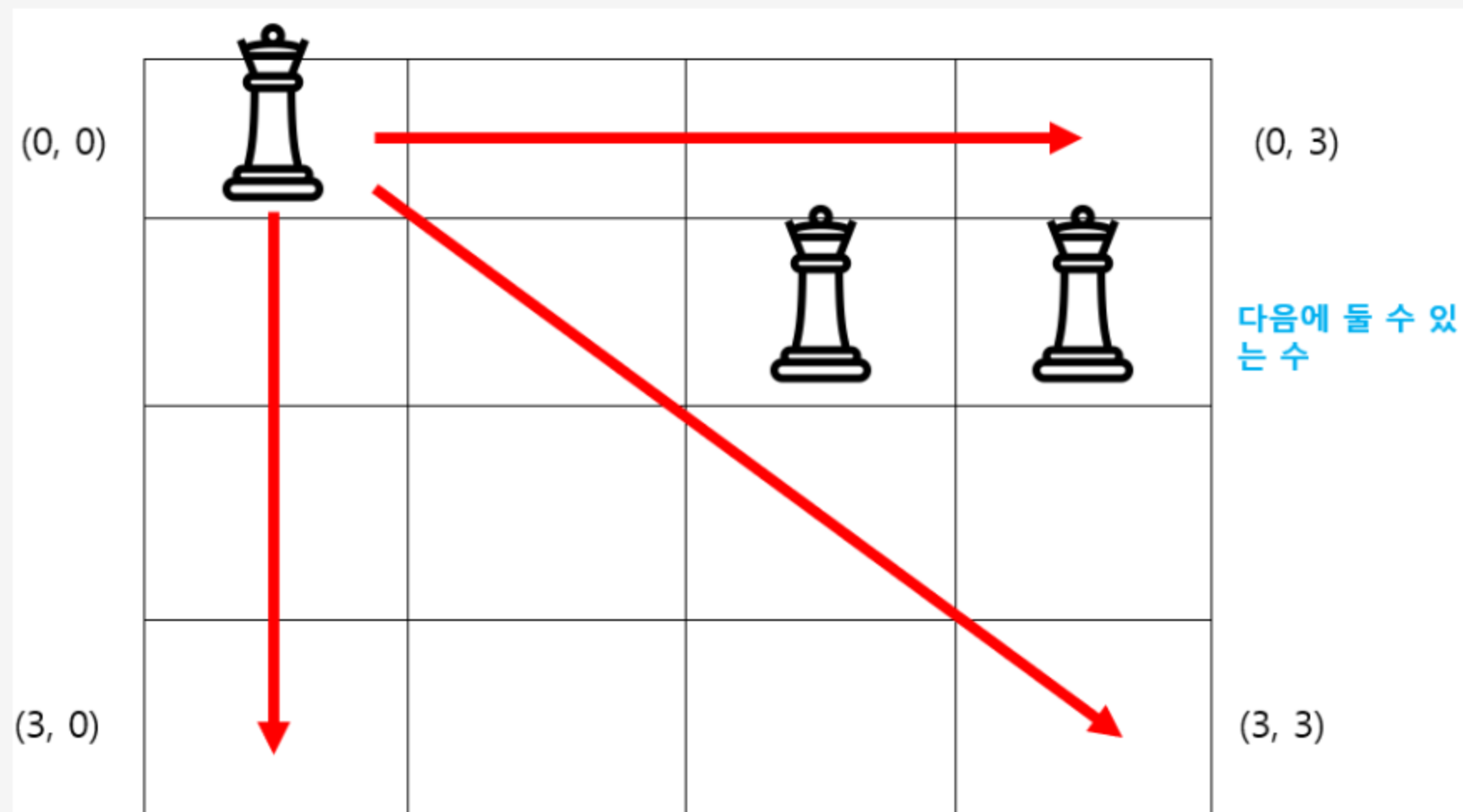
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



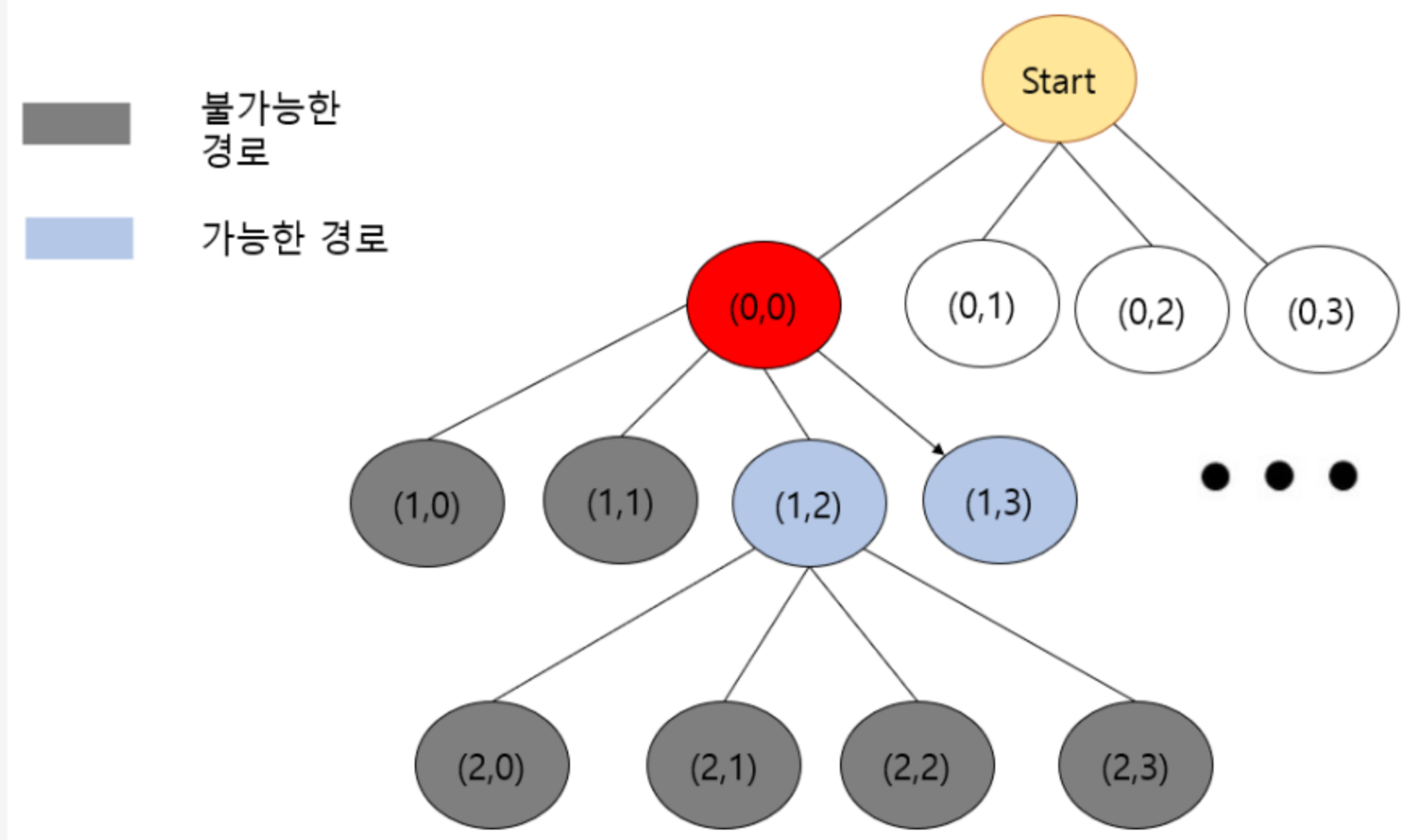
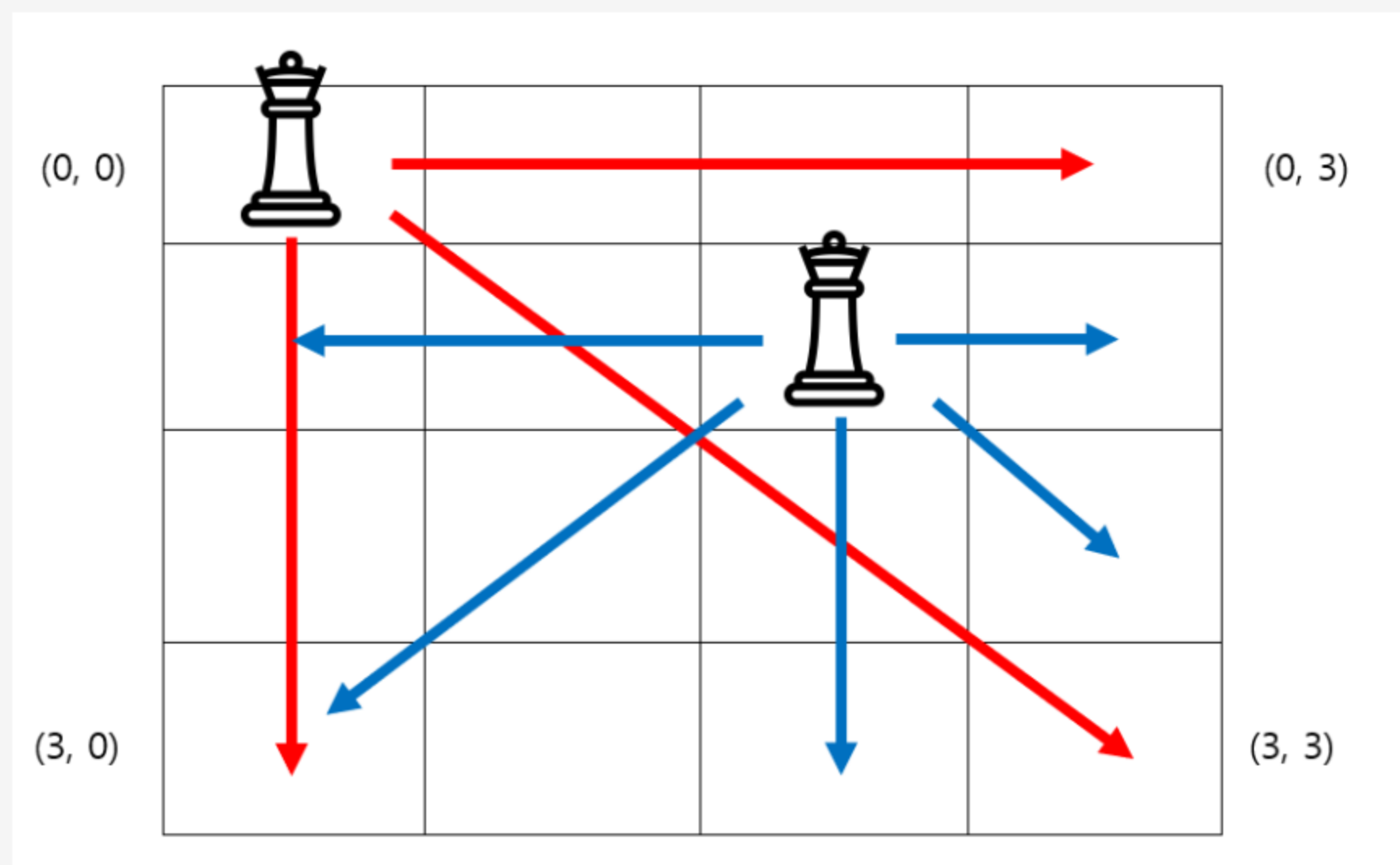
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



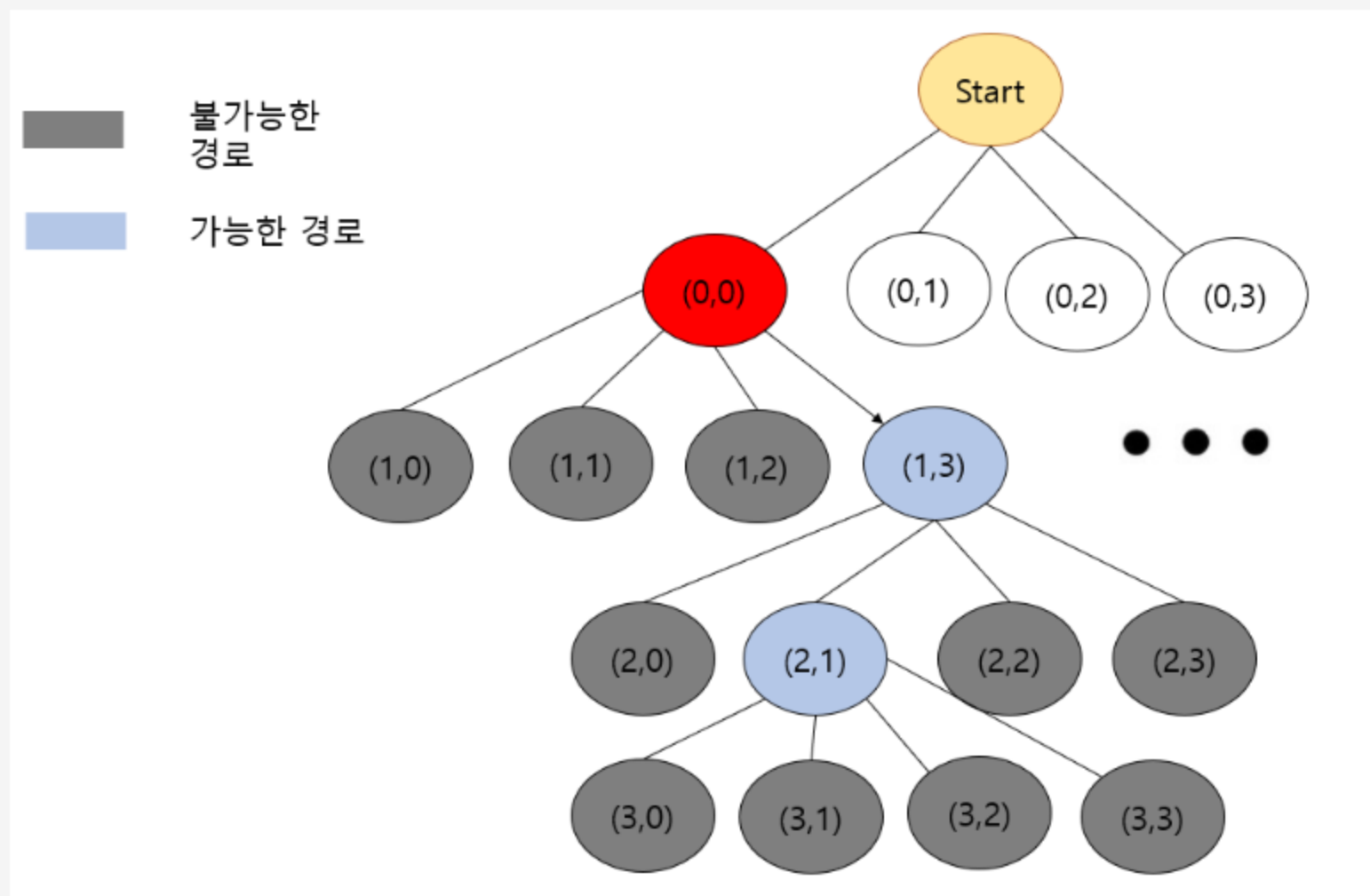
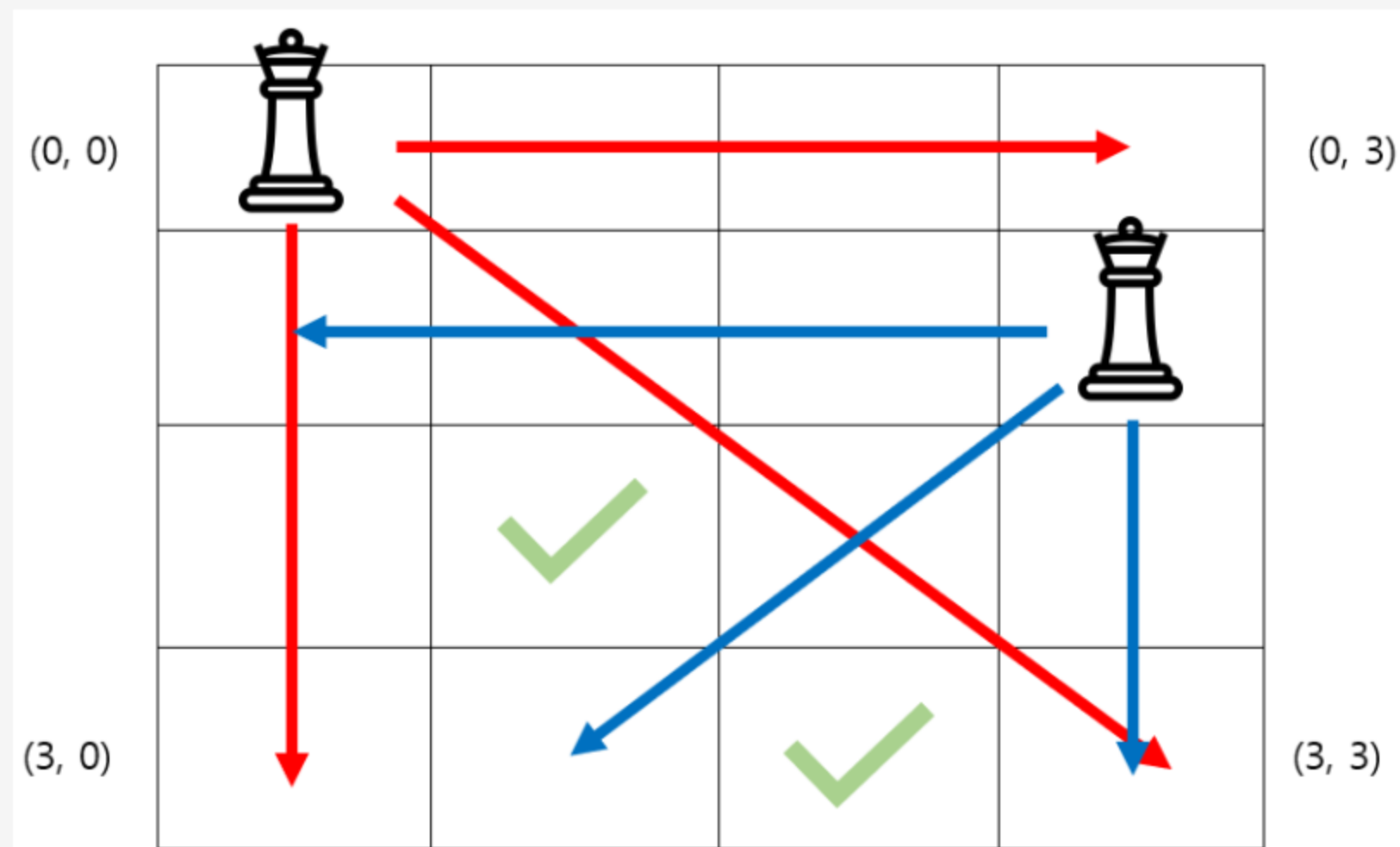
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



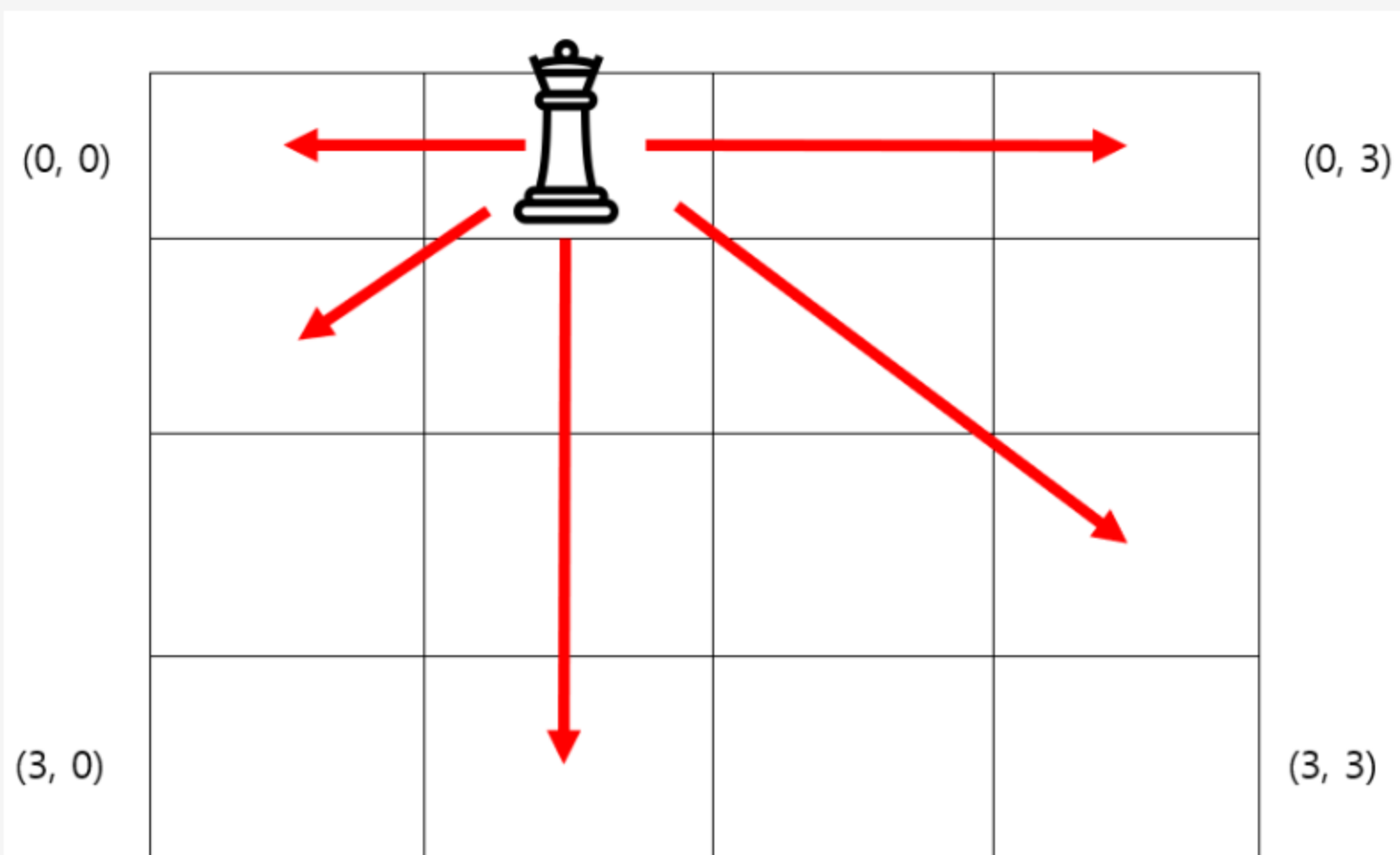
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



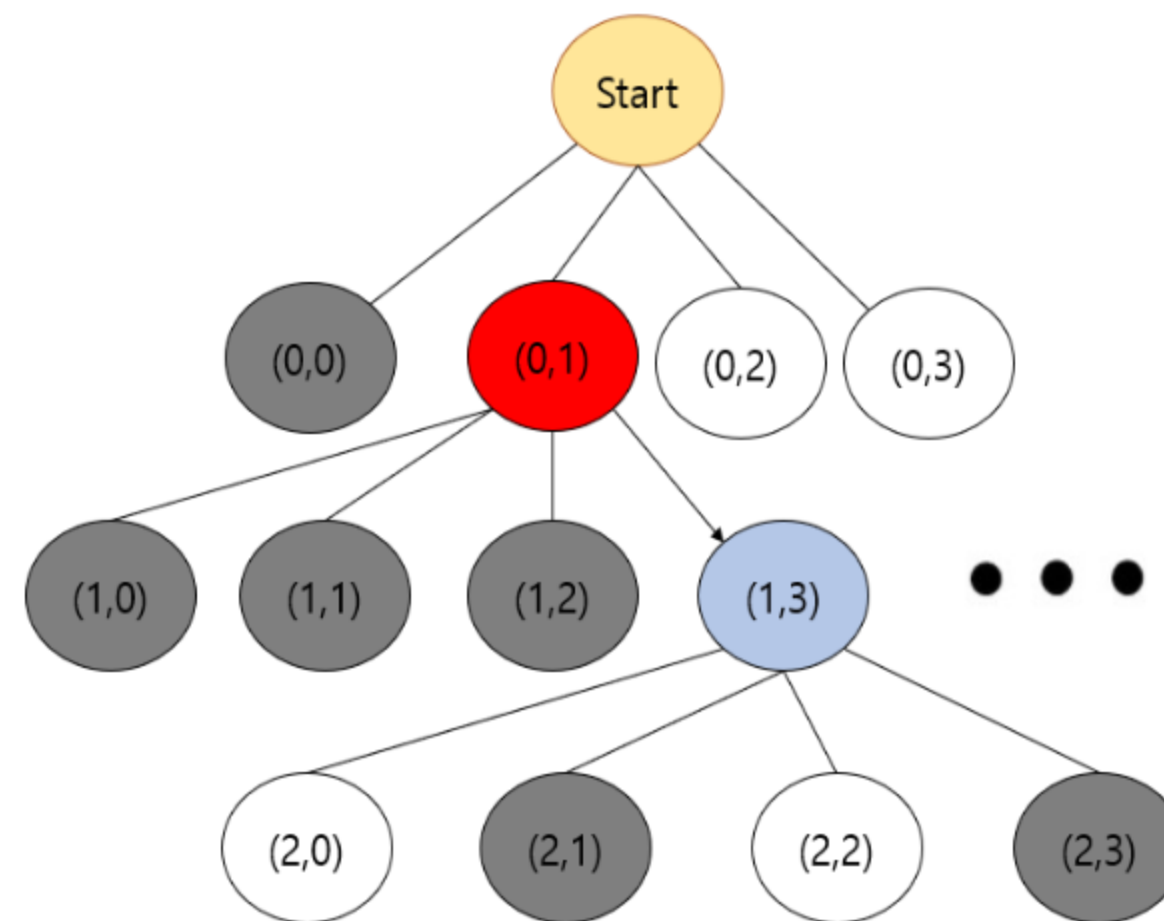
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



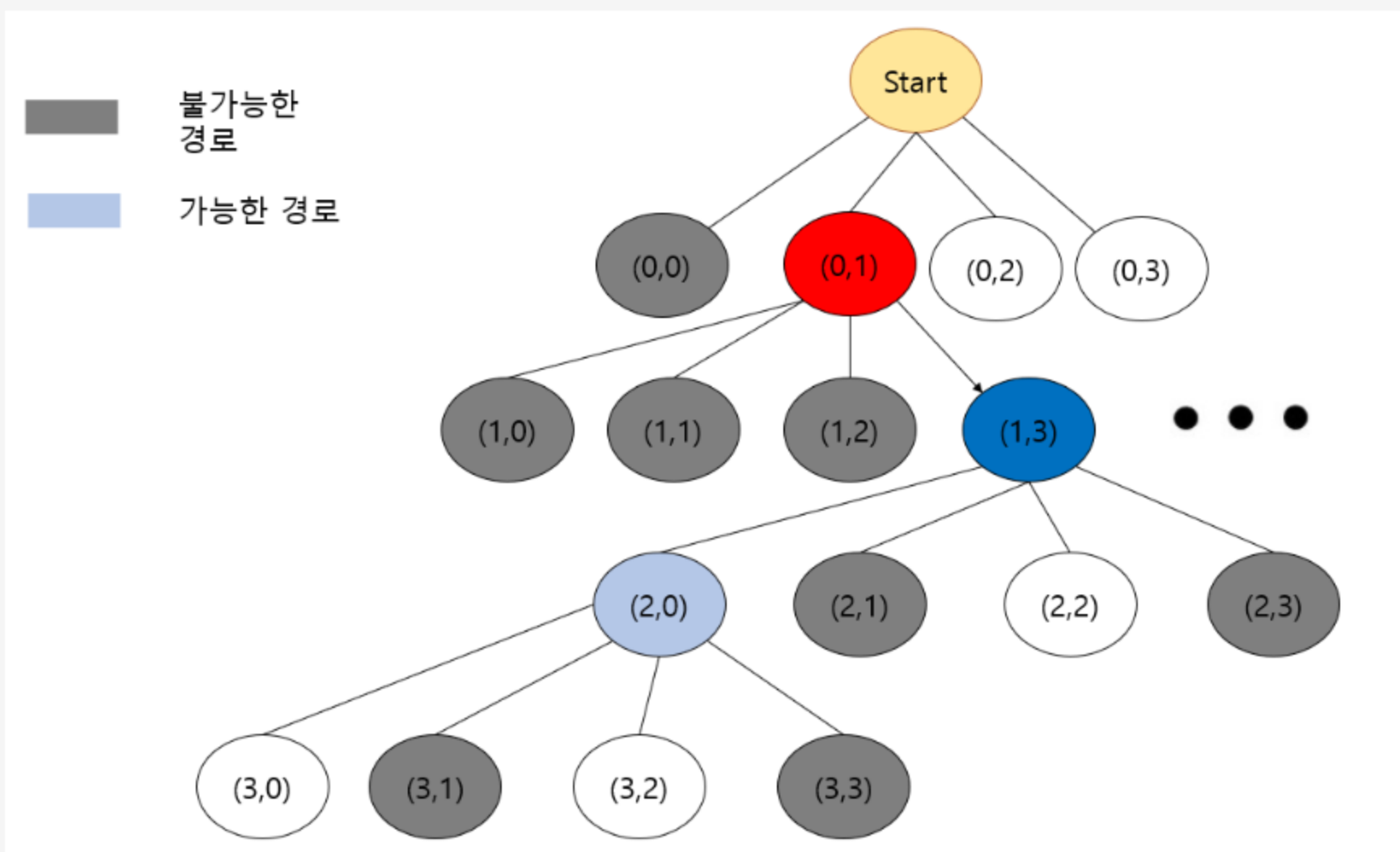
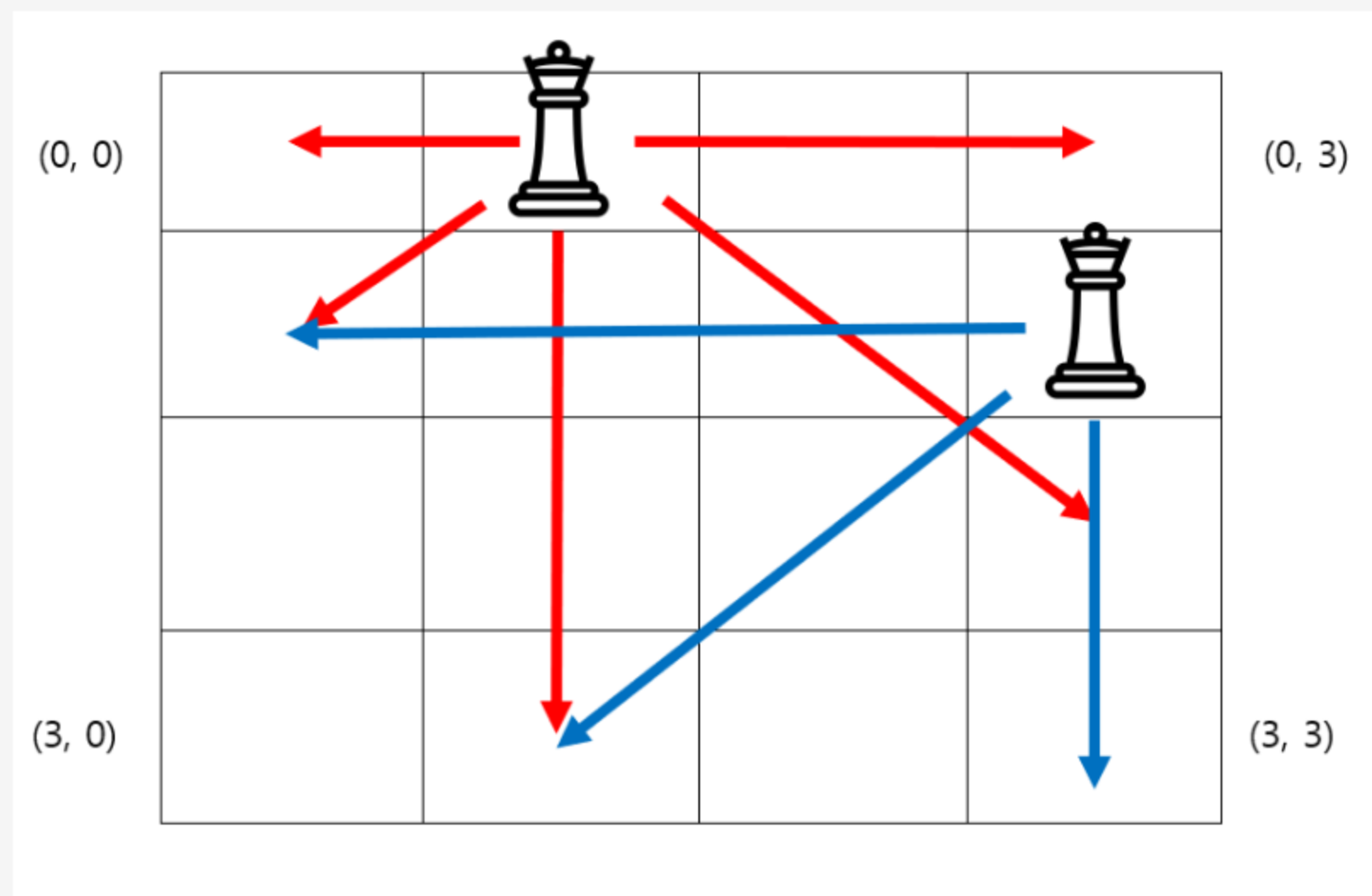
불가능한
경로

가능한 경로



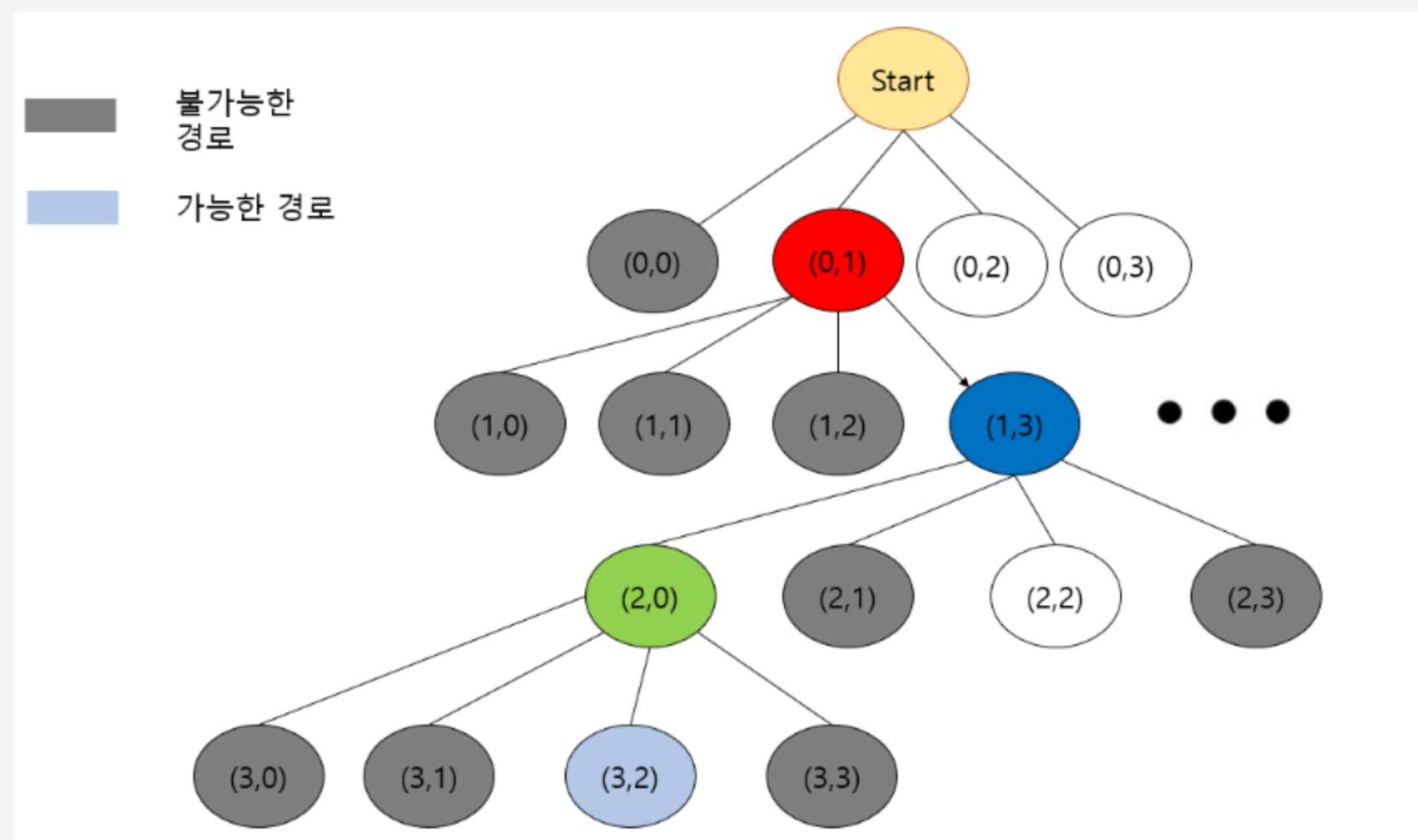
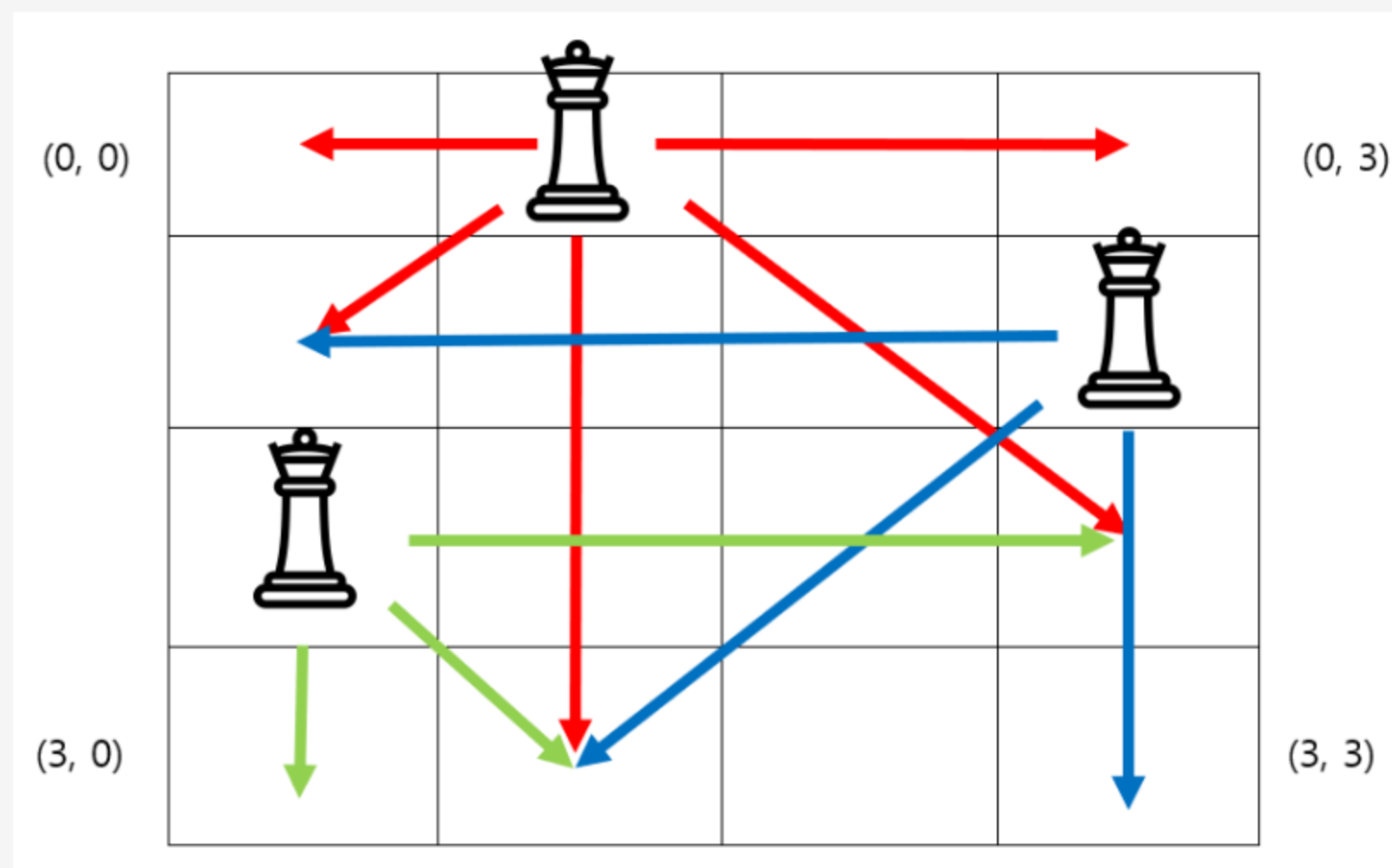
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



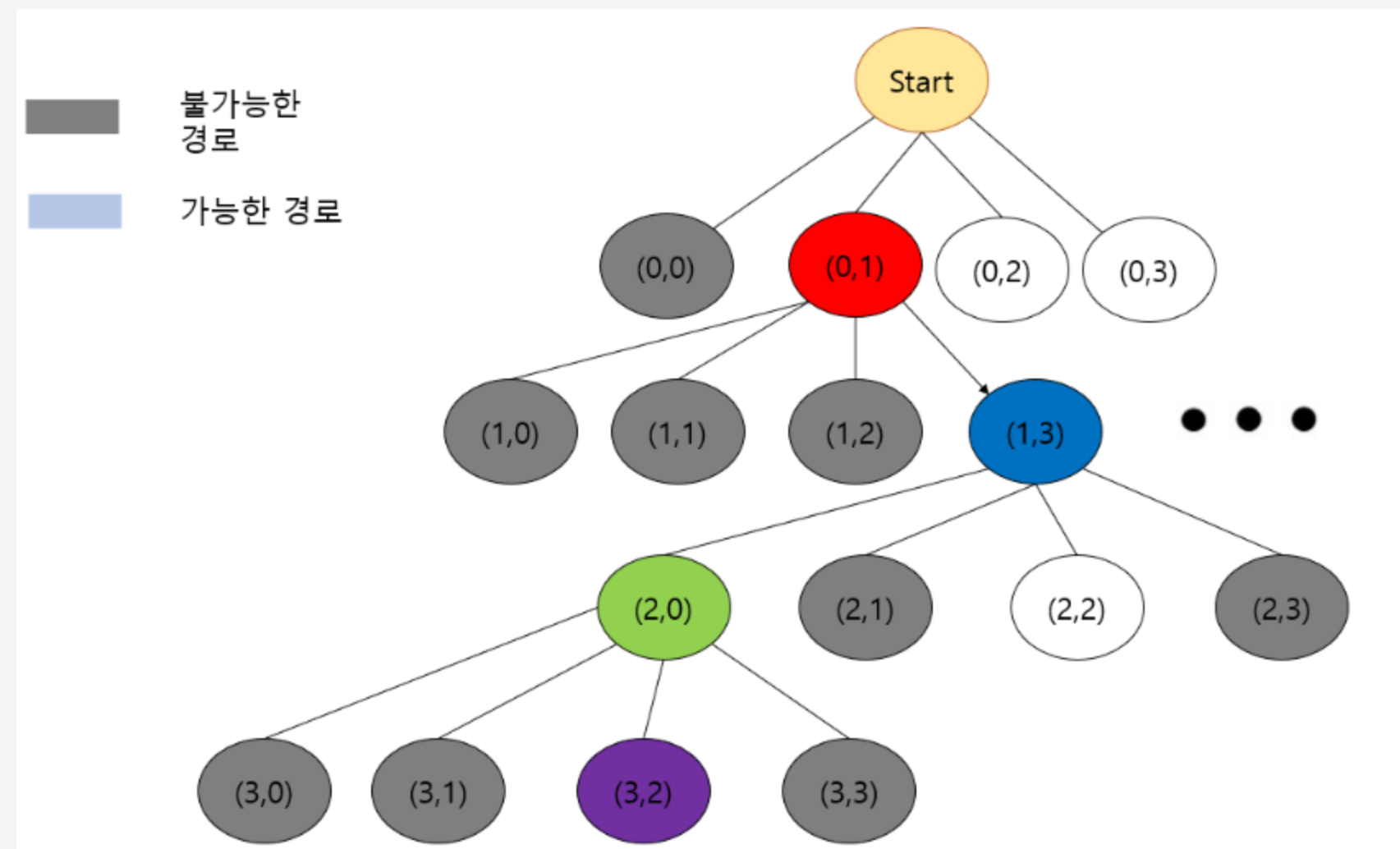
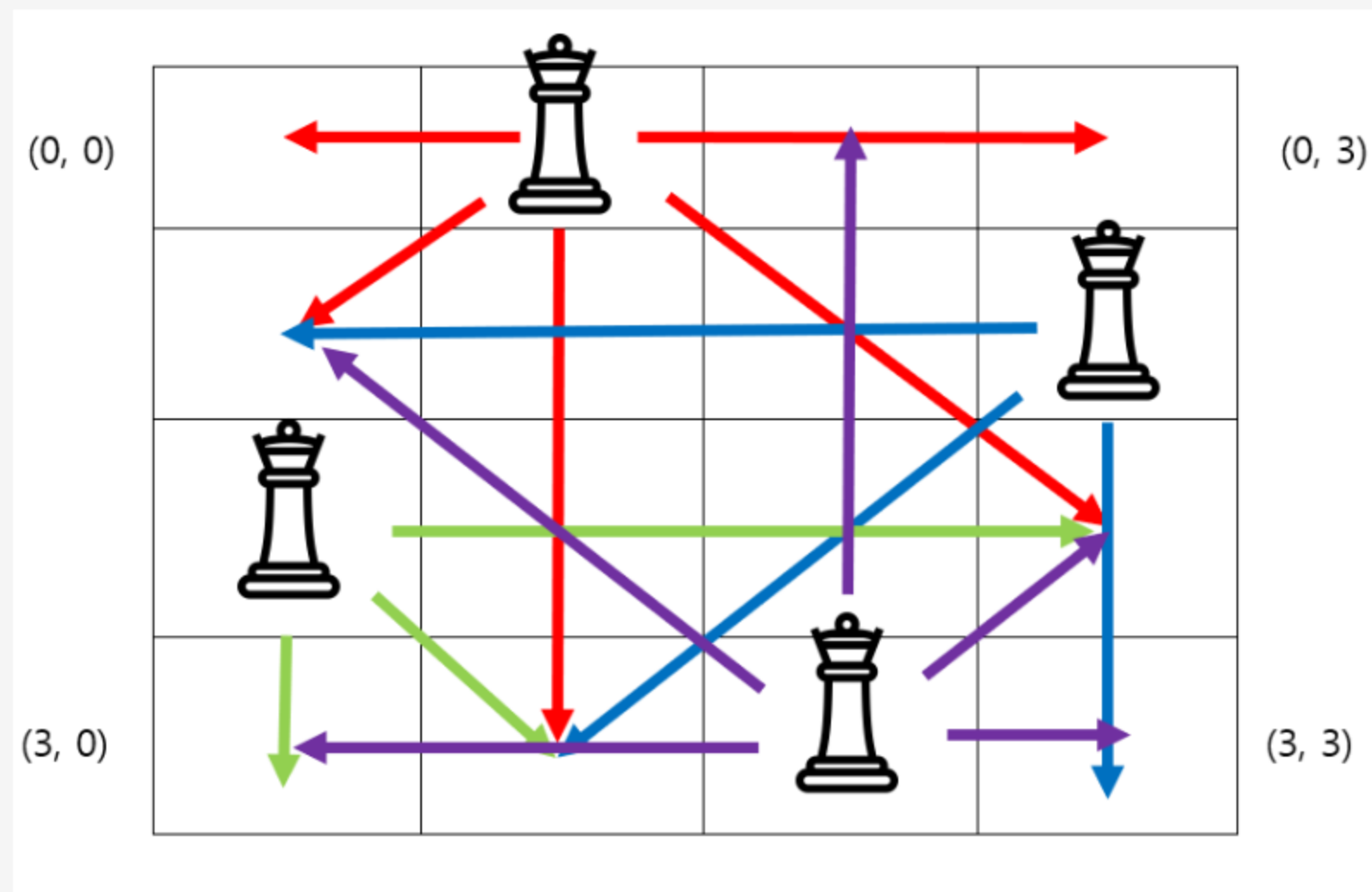
4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



4-Queen 문제

4x4 격자판에서 4개의 퀸을 놓을 수 있는가?



4-Queen 문제 - 코드

```
void go(int cnt)
{
    if (cnt == 4) {
        for (int i = 0; i < 4; i++) cout << ans[i].first << " " << ans[i].second << "\n";
        cout << "\n";
        return;
    }
    for (int i = 0; i < 4; i++) {
        if (board[cnt][i] == 0) { // 퀸을 둘 수 있는 상황
            // (cnt, i) 자리에 두었을 때, 그 다음 둘 수 없는 경우 check
            for (int k = 1; k < 4; k++) {
                //아래
                if (cnt + k < 4) board[cnt + k][i]++;
                //오른쪽 아래 대각선
                if (cnt + k < 4 && i + k < 4) board[cnt + k][i + k]++;
                //왼쪽 아래 대각선
                if (cnt + k < 4 && i - k >= 0) board[cnt + k][i - k]++;
            }
            ans[cnt] = { cnt, i };
            go(cnt + 1);
            // 백 트래킹
            for (int k = 1; k < 4; k++) {
                if (cnt + k < 4) board[cnt + k][i]--;
                if (cnt + k < 4 && i + k < 4) board[cnt + k][i + k]--;
                if (cnt + k < 4 && i - k >= 0) board[cnt + k][i - k]--;
            }
        }
    }
    return;
}
```

N-Queen 문제

BOJ 9663번 - N-Queen

N과 M (1)

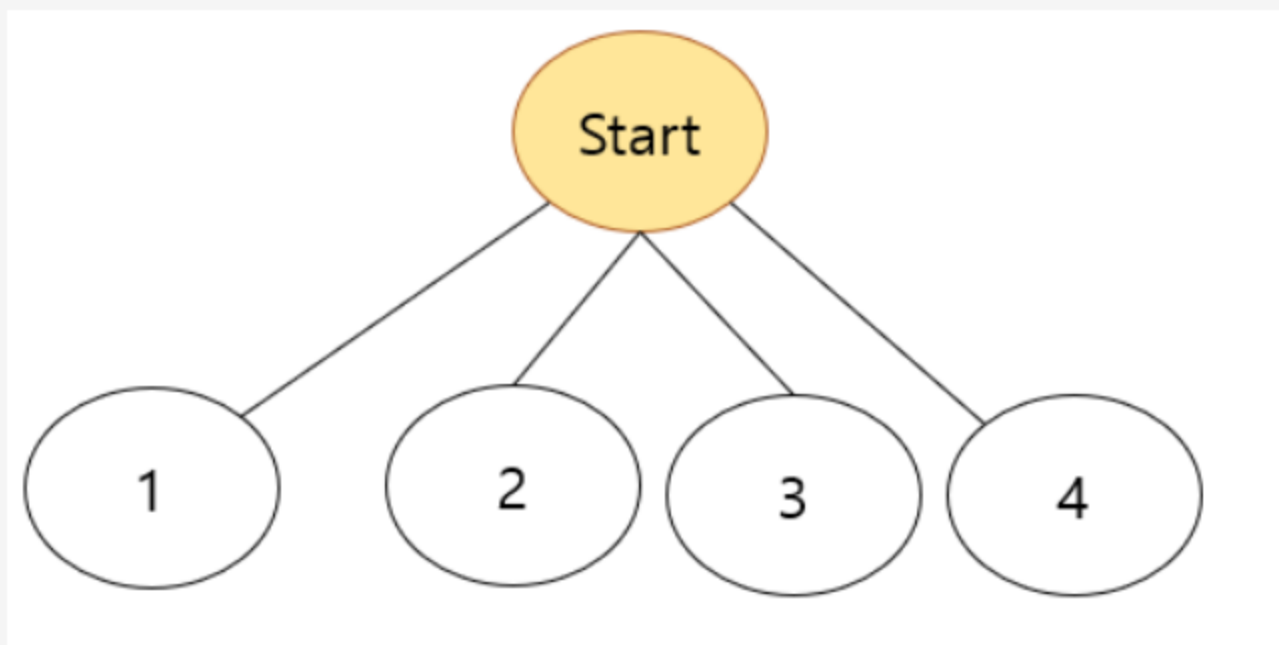
BOJ 15649번

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

N과 M (1)

$n = 4, m = 2$ $(1, 2), (1, 3), (1, 4)$
 $(2, 1), (2, 3), (2, 4)$
 $(3, 1), (3, 2), (3, 4)$
 $(4, 1), (4, 2), (4, 3)$



N과 M - 8중 for문

```
int n, m;
cin >> n >> m;
int cnt = 0;
for (int i = 1; i <= n; i++) {
    ans[cnt++] = i;
    check[i] = true;
    if (m > cnt) {
        for (int j = 1; j <= n; j++) {
            if (!check[j]) {
                ans[cnt++] = j;
                check[j] = true;
                if (m > cnt) {
                    for (int k = 1; k <= n; k++) {
                        if (!check[k]) {
                            ans[cnt++] = k;
                            check[k] = true;
                            if (m > cnt) {
                                for (int t = 1; t <= n; t++) {
                                    if (!check[t]) {
                                        ans[cnt++] = t;
                                        check[t] = true;
                                        ...
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        check[j] = false;
        cnt--;
    }
}
check[i] = false;
cnt--;
```

스타트와 링크

BOJ 14889번

모든 (i, j) 에 대한 능력치가 주어졌을 때 두 팀 능력치 차이의 최솟값을 구하는 문제

스타트와 링크

BOJ 14889번

모든 (i, j) 에 대한 능력치가 주어졌을 때 두 팀 능력치 차이의 최솟값을 구하는 문제

각 팀의 인원이 정해져 있다 $\rightarrow n / 2$ ($n \leq 20$)

= n 개 중 $n/2$ 개를 고르는 문제 (n 과 m)

스타트와 링크

BOJ 14889번

풀이 1 - n과 m 형식

n개 중 n/2 개를 고르되, 오름차순으로 골라야 한다.

순서를 지정하지 않으면 ${}_{20}C_{10}$ 으로 시간초과 발생!

시간 복잡도 : $O(2^N)$

스타트와 링크

BOJ 14889번

풀이 2 - 양자택일 방법

1 ~ n번째 사람마다 한 번은 스타트 팀, 한 번은 링크 팀에 들어가 본다.

기저 사례 처리가 좀 더 간편

시간 복잡도 : $O(2^N)$

부등호

BOJ 2529번

k개의 부등호 순서를 만족하는 (k+1)자리의 정수 중에서 최댓값과 최솟값을 찾는 문제

선택된 숫자는 모두 달라야 한다!

부등호

BOJ 2529번

조건 check!

k의 범위 ($2 \leq k \leq 9$)

들어가는 숫자는 0 ~ 9 (10가지)

모든 경우를 확인? -> 10!

부등호

BOJ 2529번

풀이 1- 백트래킹

백트래킹 방법을 이용

부등호에 따라 다음에 나올 수 없는 수를 거를 수 있다!

부등호

BOJ 2529번

풀이 2- 순열

최댓값은 987.. 부터 시작해 prev_permutation

최솟값은 012.. 부터 next_permutation

만들어 놓고 차례대로 비교!