2021 겨울 신촌 연합 알고리즘 캠프

완전탐색 & 백트래킹

초급 알고리즘

HI-ARC 김기선

완전탐색(Brute Force)이란?

- 가능한 경우의 수를 모두 조사한다.

Ex) 번호 자물쇠

최악의 경우 10000 * 1초 = 대략 166분

컴퓨터의 연산 속도 : 대략 1초에 1억

백준 2231 분해합

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	192 MB	36127	17587	14232	48.334%

문제

어떤 자연수 N이 있을 때, 그 자연수 N의 분해합은 N과 N을 이루는 각 자리수의 합을 의미한다. 어떤 자연수 M의 분해합이 N인 경우, M을 N의 생성자라 한다. 예를 들어, 245의 분해합은 256(=245+2+4+5)이 된다. 따라서 245는 256의 생성자가 된다. 물론, 어떤 자연수의 경우에는 생성자가 없을 수도 있다. 반대로, 생성자가 여러 개인 자연수도 있을 수 있다.

자연수 N이 주어졌을 때, N의 가장 작은 생성자를 구해내는 프로그램을 작성하시오.

입력

첫째 줄에 자연수 N(1 ≤ N ≤ 1,000,000)이 주어진다.

출력

첫째 줄에 답을 출력한다. 생성자가 없는 경우에는 0을 출력한다.

예제 입력 1 복사

```
216
```

예제 출력 1 복사

```
198
```

백준 1018 체스판 다시 칠하기

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	30093	13598	11247	46.301%

문제

지민이는 자신의 저택에서 MN개의 단위 정사각형으로 나누어져 있는 M*N 크기의 보드를 찾았다. 어떤 정사각형은 검은색으로 칠해져 있고, 나머지는 흰색으로 칠해져 있다. 지민이는 이 보드를 잘라서 8*8 크기의 체스판으로 만들려고 한다.

체스판은 검은색과 흰색이 번갈아서 칠해져 있어야 한다. 구체적으로, 각 칸이 검은색과 흰색 중 하나로 색칠되어 있고, 변을 공유하는 두 개의 사각형은 다른 색으로 칠해져 있어야 한다. 따라서 이 정의를 따르면 체스판을 색칠하는 경우는 두 가지뿐이다. 하나는 맨 왼쪽 위 칸이 흰색인 경우, 하나는 검은색인 경우이다.

보드가 체스판처럼 칠해져 있다는 보장이 없어서, 지민이는 8*8 크기의 체스판으로 잘라낸 후에 몇 개의 정사각형을 다시 칠해야겠다고 생각했다. 당연히 8*8 크기는 아무데서나 골라도 된다. 지민이가 다시 칠해야 하는 정사각형의 최소 개수를 구하는 프로그램을 작성하시오.

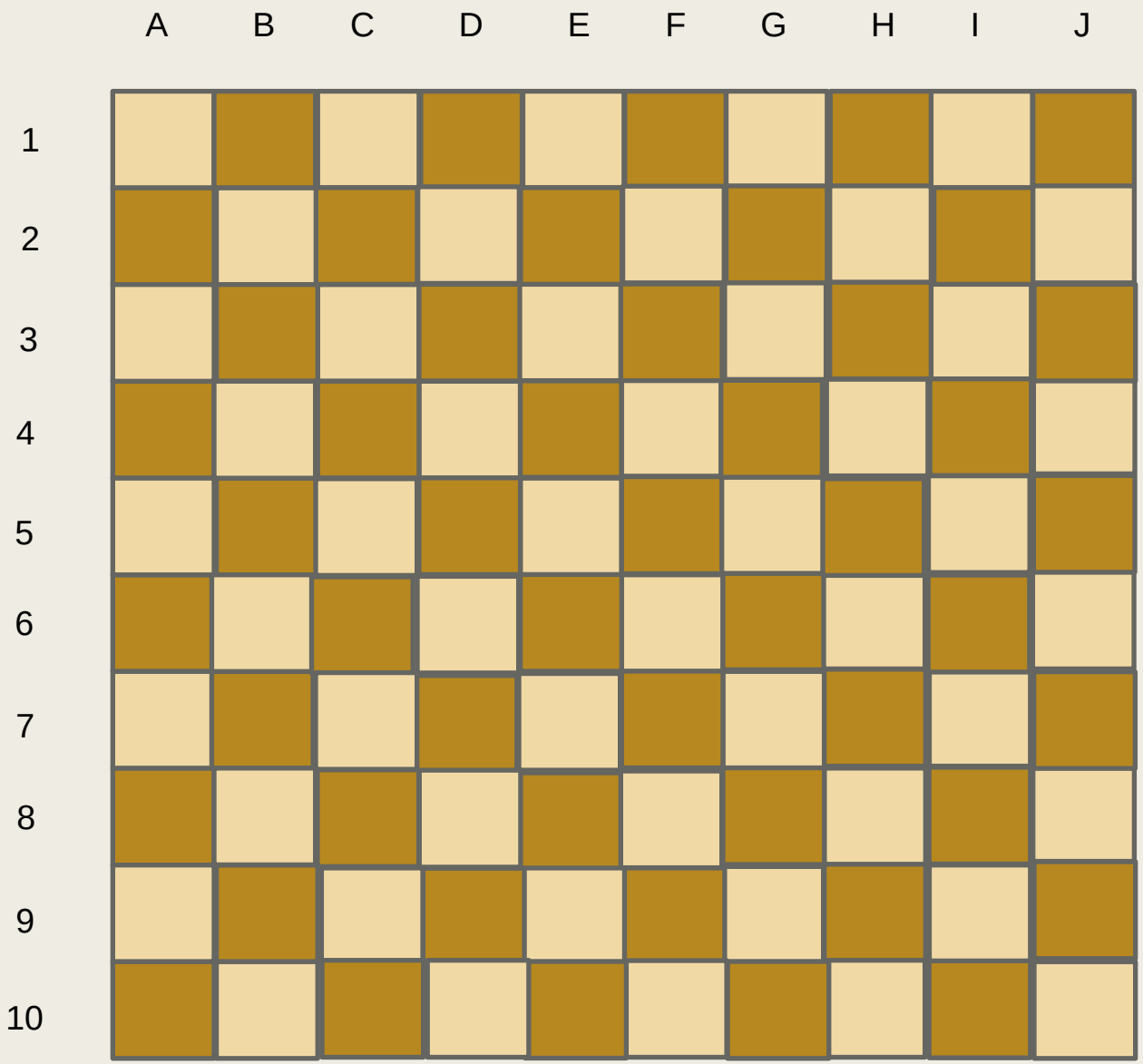
입력

첫째 줄에 N과 M이 주어진다. N과 M은 8보다 크거나 같고, 50보다 작거나 같은 자연수이다. 둘째 줄부터 N개의 줄에는 보드의 각 행의 상태가 주어진다. B는 검은색이며, W는 흰색이다.

출력

첫째 줄에 지민이가 다시 칠해야 하는 정사각형 개수의 최솟값을 출력한다.

1018 체스판 다시 칠하기

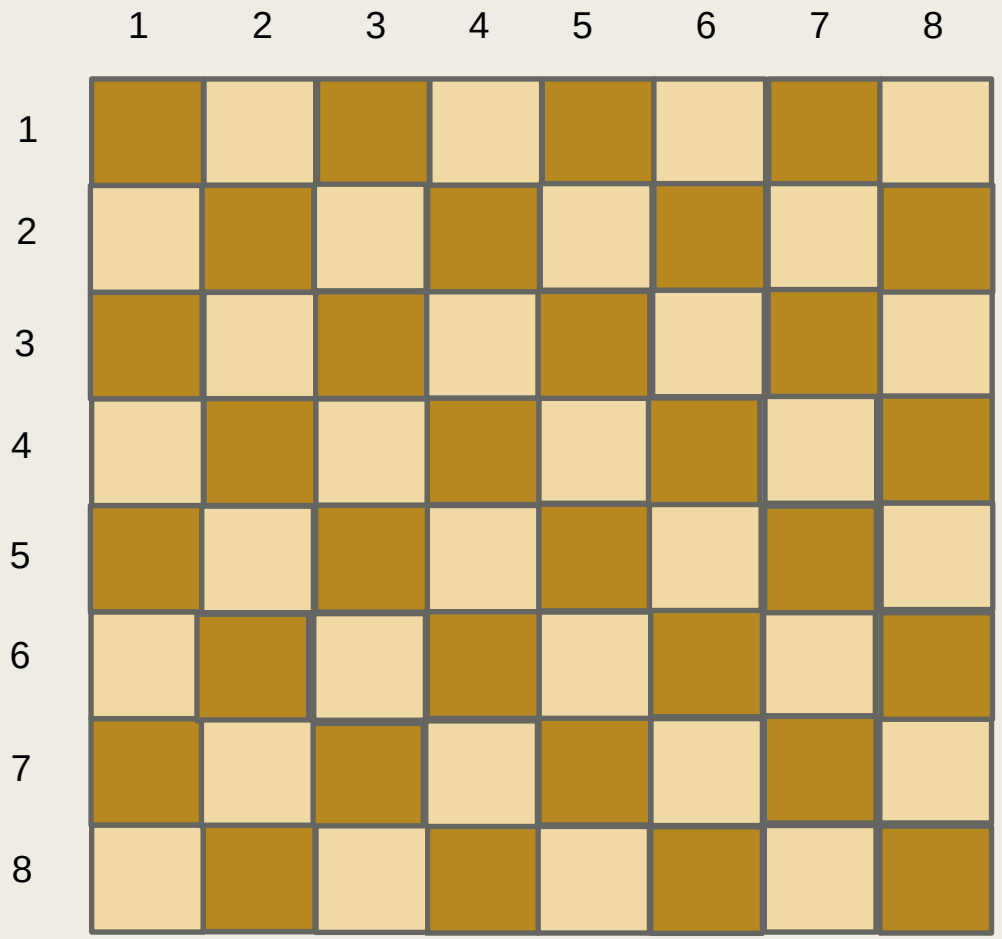


체스판의 이름을 정해야 한다.

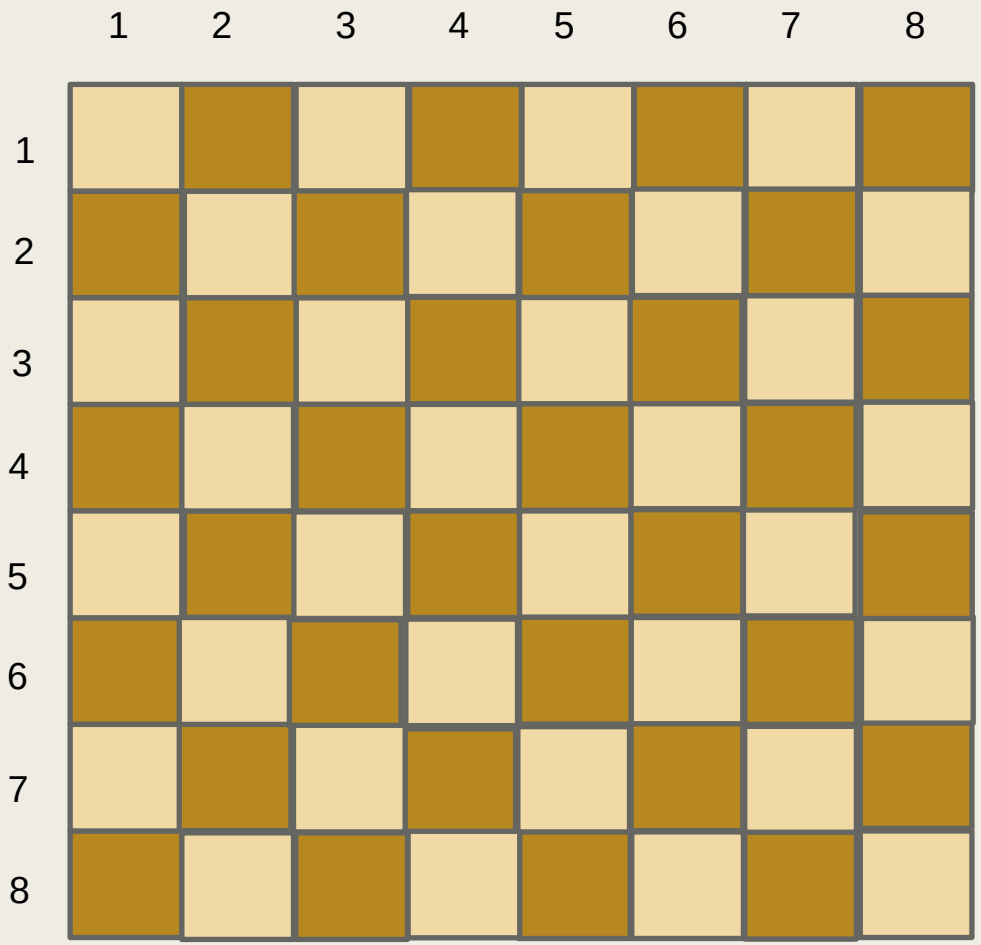
-> index로

1018 체스판 다시 칠하기

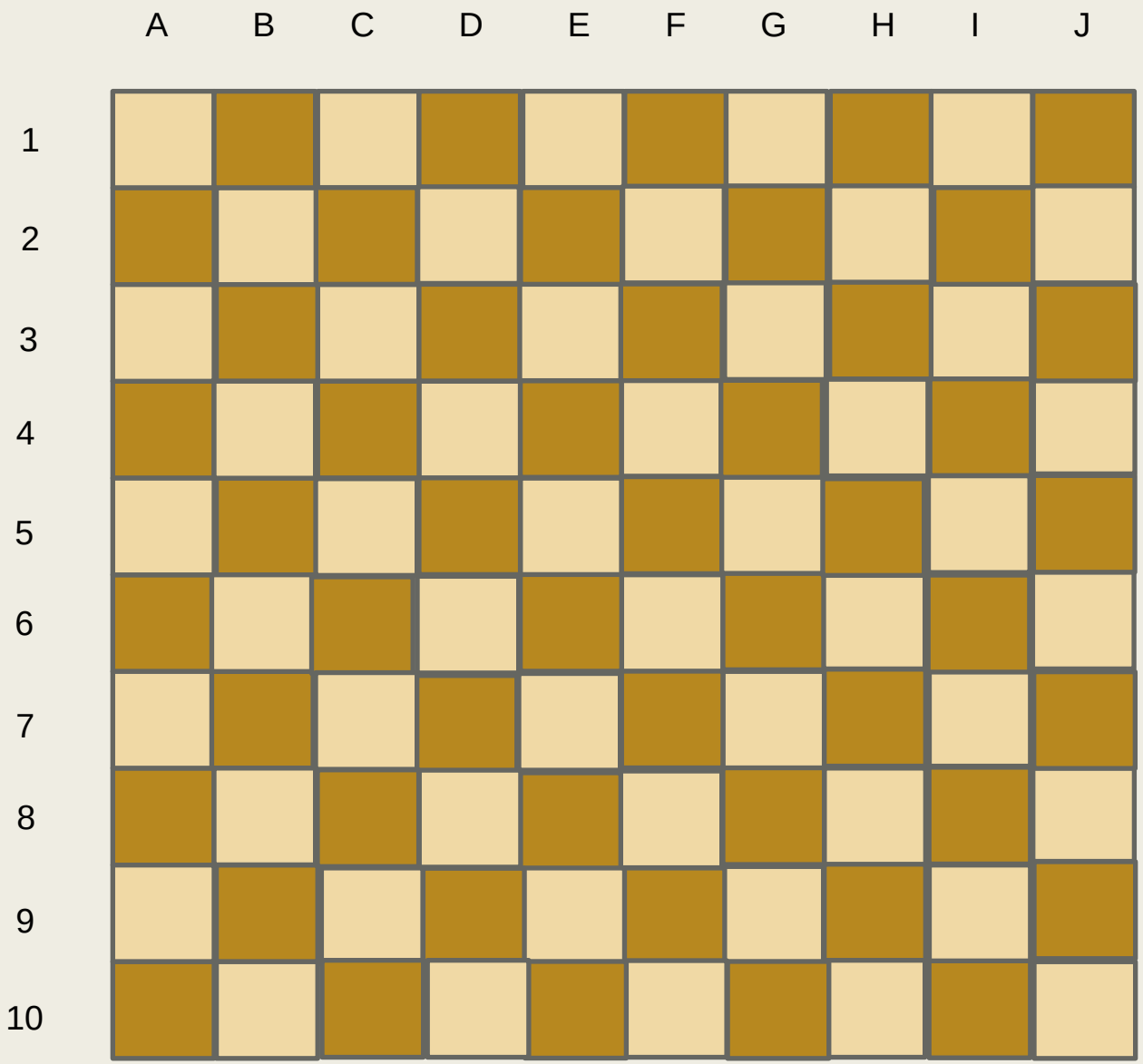
왼쪽 위가 검은색인 경우 (B 체스판)



왼쪽 위가 흰색인 경우 (W 체스판)



1018 체스판 다시 칠하기



1018 체스판 다시 칠하기

따라서 반드시 두가지 케이스

Case 1 : 모든 B 홀수이면서 모든 W 짝수
Case 2 : 모든 B 짝수이면서 모든 W 홀수

```
for (int a = 0; a <= n - 8; a++) {
    for (int b = 0; b <= m - 8; b++) {
        int num1 = 0, num2 = 0;
        for (int i = a; i < a + 8; i++) {
            for (int j = b; j < b + 8; j++) {
                if (mat[i][j] == 'W') {
                    if ((i + j) % 2 == 0)num1++;
                    else num2++;
                }
                else {
                    if ((i + j) % 2 == 0)num2++;
                    else num1++;
                }
            }
        }
        ans = min({ ans, num1, num2 });
    }
}
```

백준 15649 N과 M (1)

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	26064	15785	10692	60.237%

문제

자연수 N과 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구하는 프로그램을 작성하시오.

- 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열

입력

첫째 줄에 자연수 N과 M이 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 복사

```
3 1
```

예제 출력 1 복사

```
1
2
3
```

재귀함수란?

```
void dream(int dream_level) {
    dream(dream_level + 1);
}
```

어떤 함수에서 자기 자신을 호출하는 함수
재귀함수는 인셉션에서의 꿈과 동일하다고 보면 된다.

이 코드의 문제점은? 기저조건(탈출조건)을 잘 세우자!

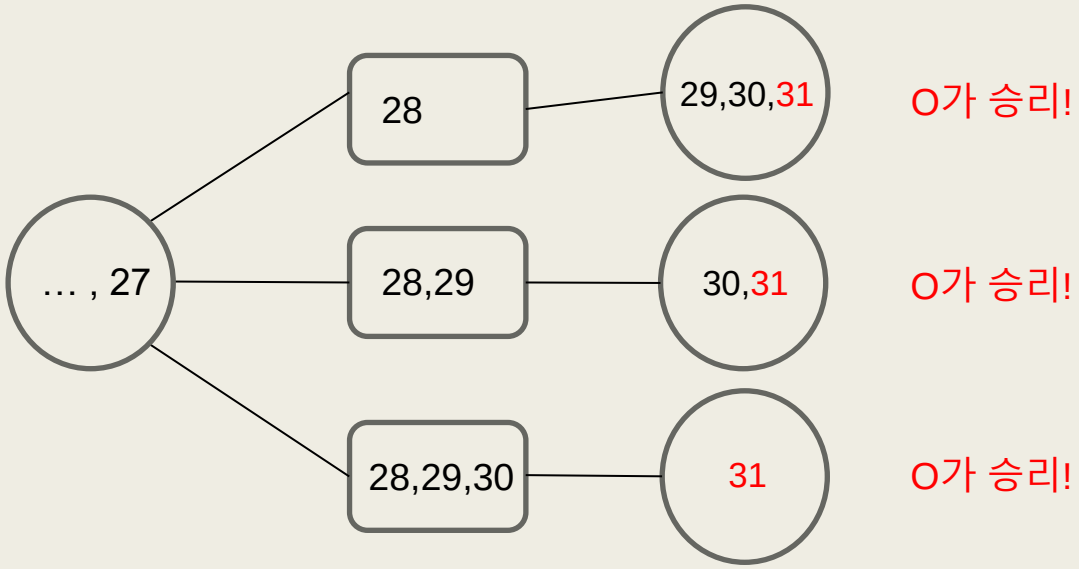
```
void dream(int dream_level) {
    int a = 1;
    if (a == 4)return;
    dream(dream_level + 1);
    a++;
}
```

- 변수를 알맞게 사용하여 값들을 잘 관리하자

Backtracking

- 현재 상태에서 가능한 후보군으로 가지를 치며 탐색하는 알고리즘

예시) 베스킨라빈스 31게임 (31을 외치면 이긴다)



N과 M (1) 중복 X

N=5, M=3

N과 M (1)

```cpp
#include<iostream>
using namespace std;
#define MAX 10

int N, M;
int answer[MAX]; // 사용한 숫자들을 저장해서 출력용
bool isused[MAX]; // 사용한 숫자 여부 확인용

void solve(int level) {
    if (level == M) {
        for (int i = 0; i < M; i++)cout << answer[i] << " ";
        cout << '\n';
        return;
    }
    for (int i = 1; i <= N; i++) {
        if (isused[i])continue;
        isused[i] = true;
        answer[level] = i;
        solve(level + 1);
        isused[i] = false;
    }
}

int main() {
    cin >> N >> M;
    solve(0);
}
```

백준 15650 N과 M (2)

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	17015	12951	9482	75.686%

문제

자연수 N과 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구하는 프로그램을 작성하시오.

- 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열
- 고른 수열은 오름차순이어야 한다.

입력

첫째 줄에 자연수 N과 M이 주어진다. ($1 \le M \le N \le 8$)

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.

N과 M (2)

중복 X, 오름차순

N=5, M=3

N과 M (2)

```cpp
#include<iostream>
using namespace std;
#define MAX 10

int N, M;
int answer[MAX]; // 사용한 숫자들을 저장해서 출력용

void solve(int level, int num) {
    if (level == M) {
        for (int i = 0; i < M; i++)cout << answer[i] << " ";
        cout << '\n';
        return;
    }
    for (int i = num; i <= N; i++) {
        answer[level] = i;
        solve(level + 1, i+1);
    }
}

int main() {
    cin >> N >> M;
    solve(0, 1);
}
```

백준 9663 N-Queen

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
10 초	128 MB	31473	17041	11196	53.793%

문제

N-Queen 문제는 크기가 N × N인 체스판 위에 퀸 N개를 서로 공격할 수 없게 놓는 문제이다.

N이 주어졌을 때, 퀸을 놓는 방법의 수를 구하는 프로그램을 작성하시오.

입력

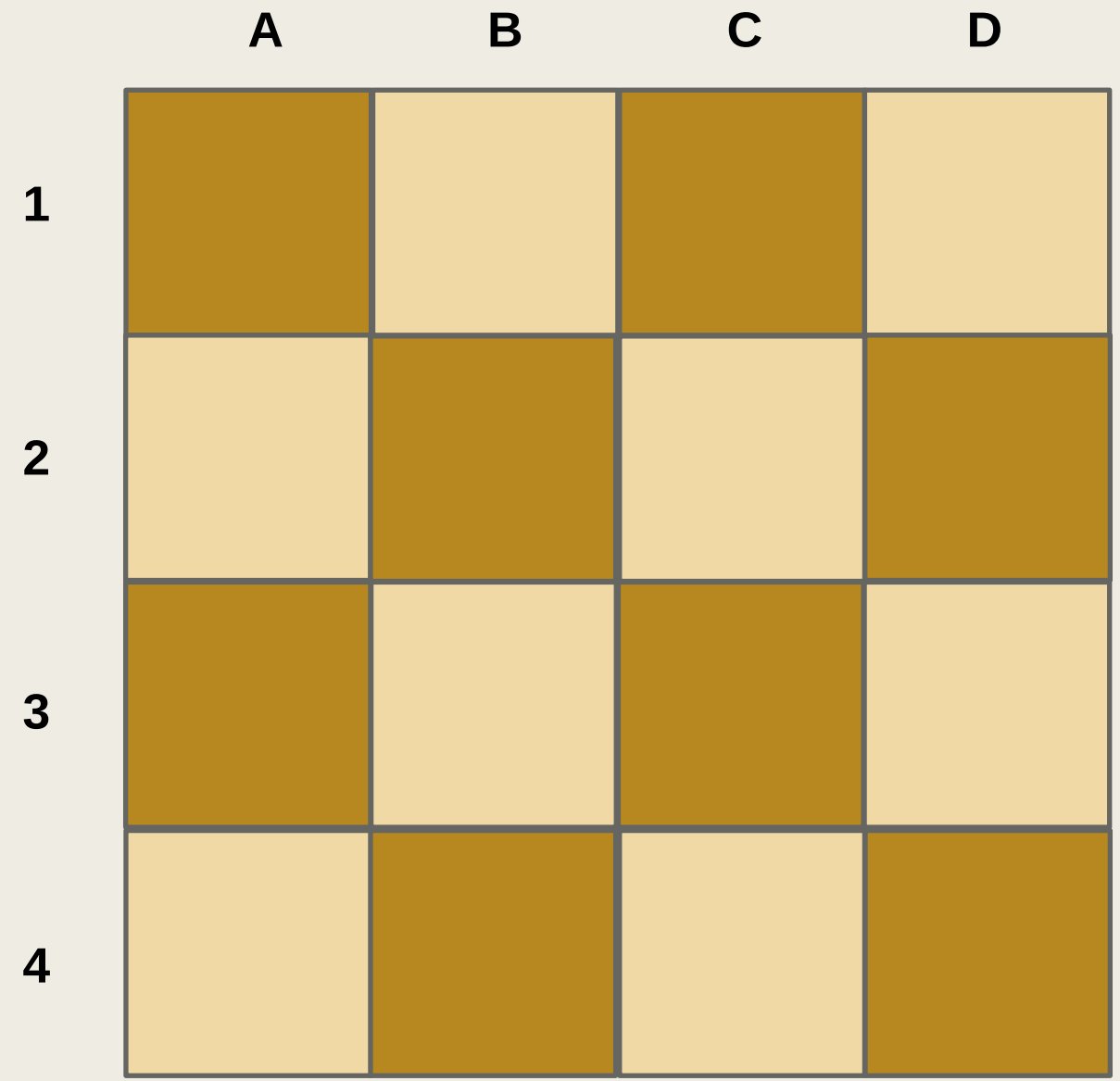
첫째 줄에 N이 주어진다. ($1 \le N < 15$)

출력

첫째 줄에 퀸 N개를 서로 공격할 수 없게 놓는 경우의 수를 출력한다.

9663 N-Queen

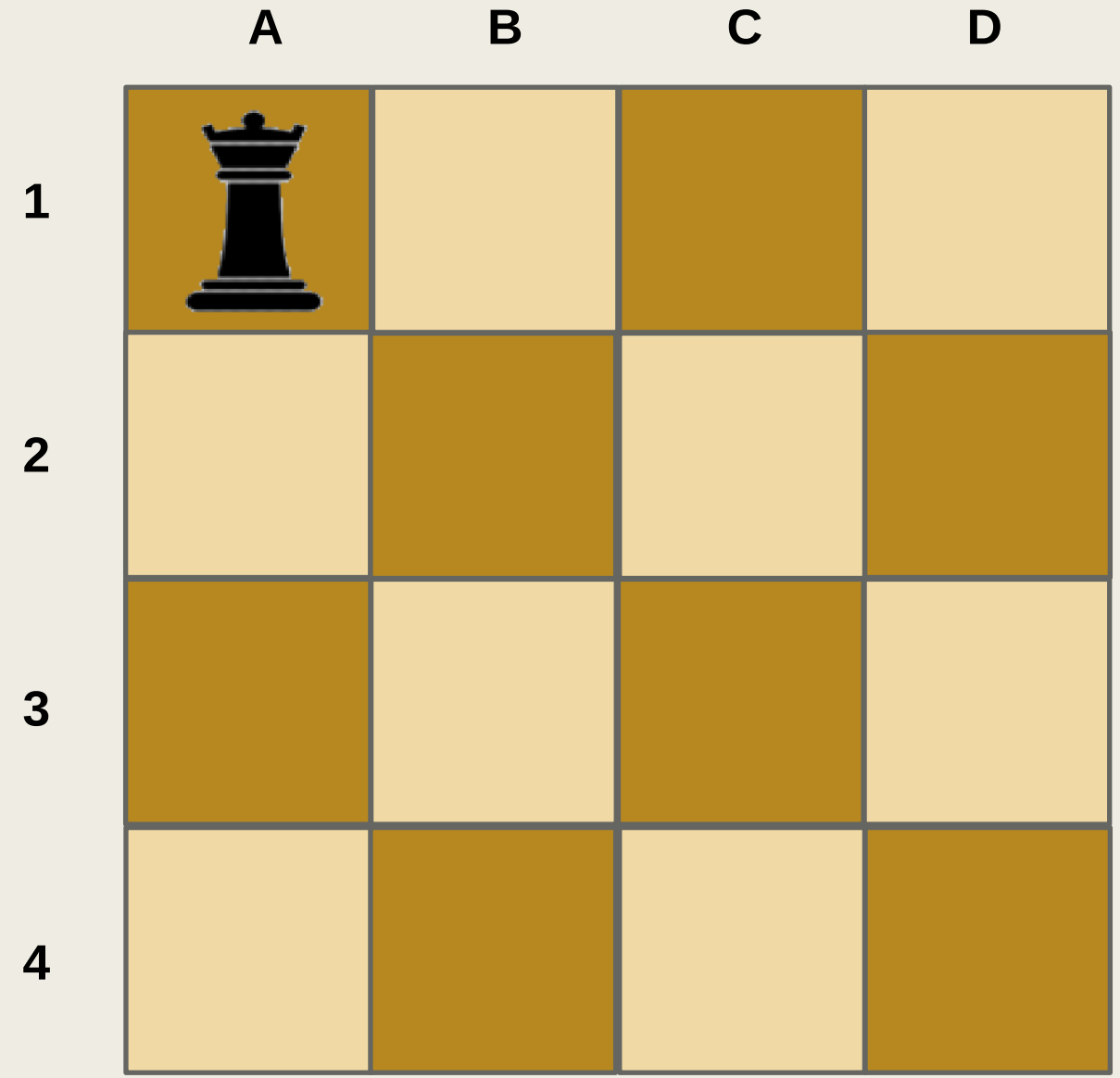
N = 4



A1 ~ D4 후보군 중 A1에 둬보자

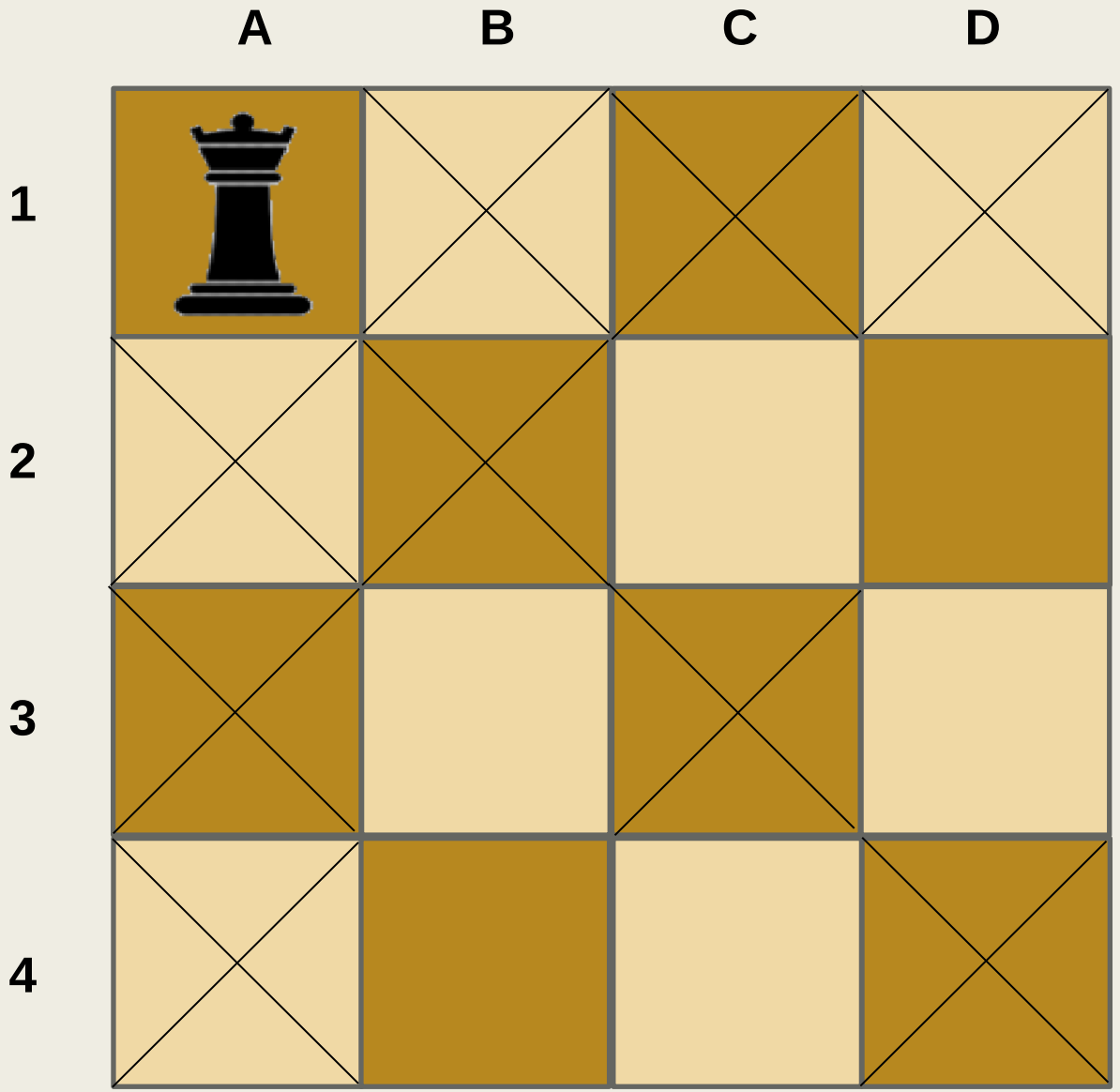
9663 N-Queen

N = 4



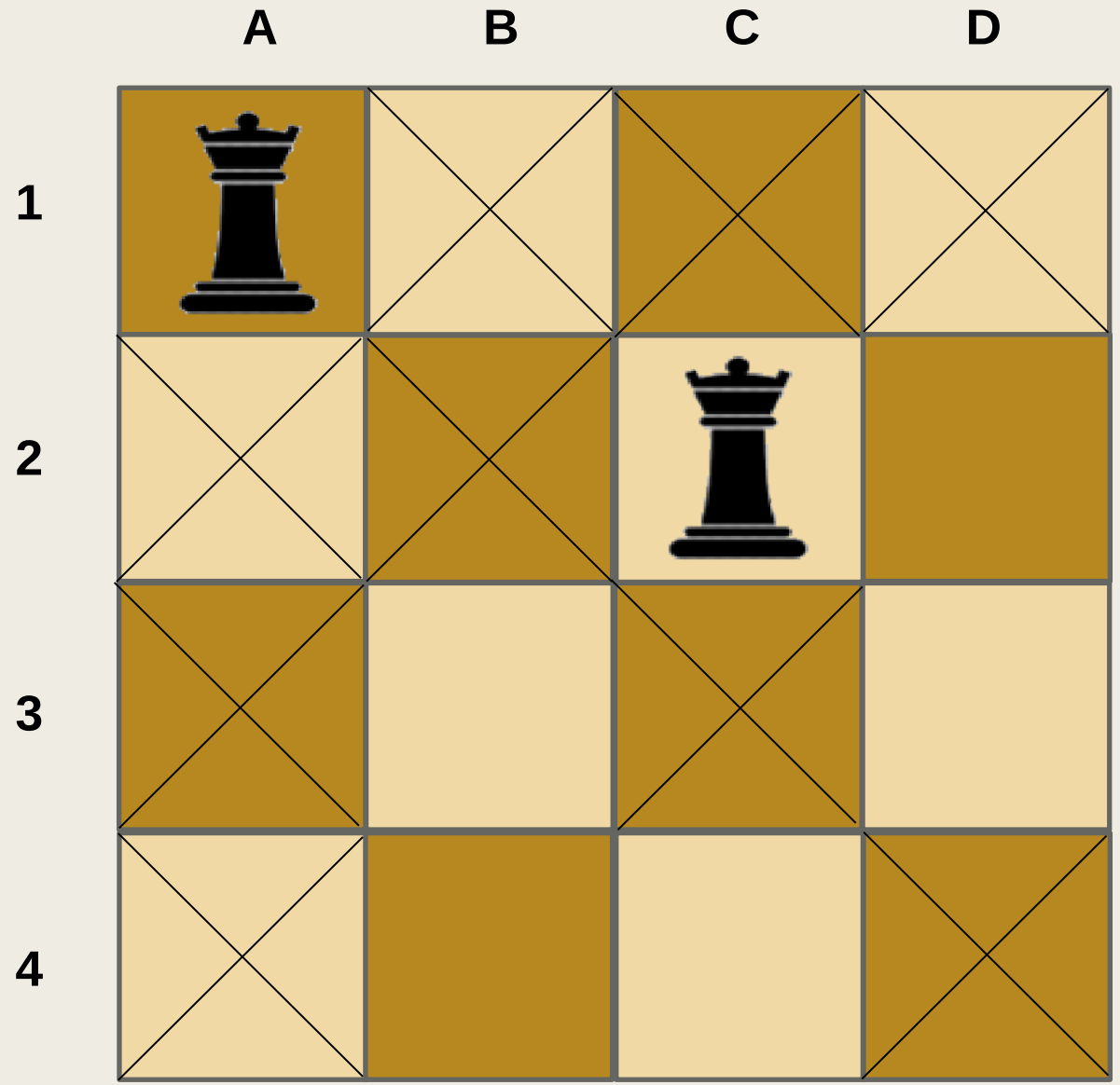
9663 N-Queen

N = 4



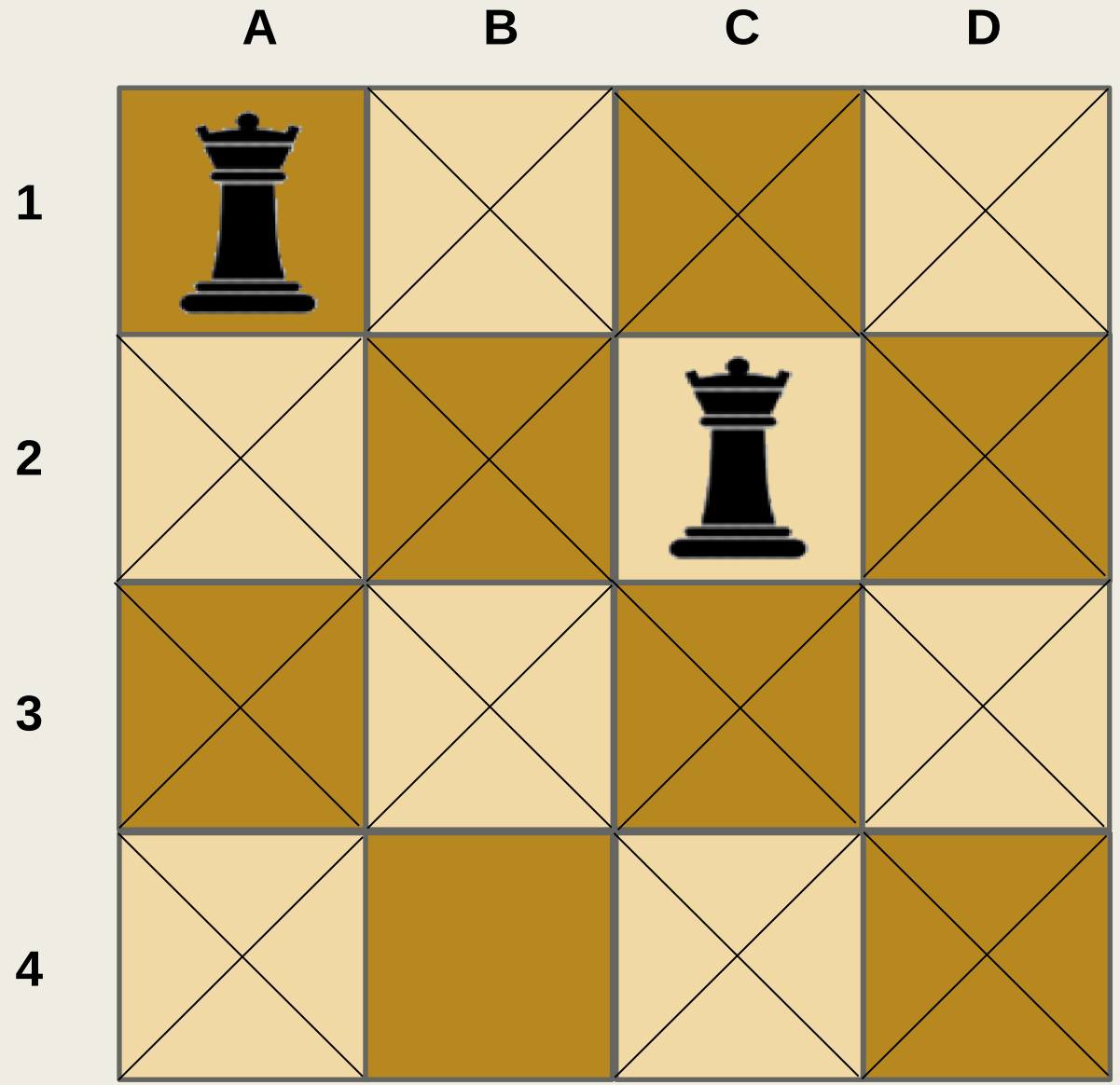
9663 N-Queen

N = 4



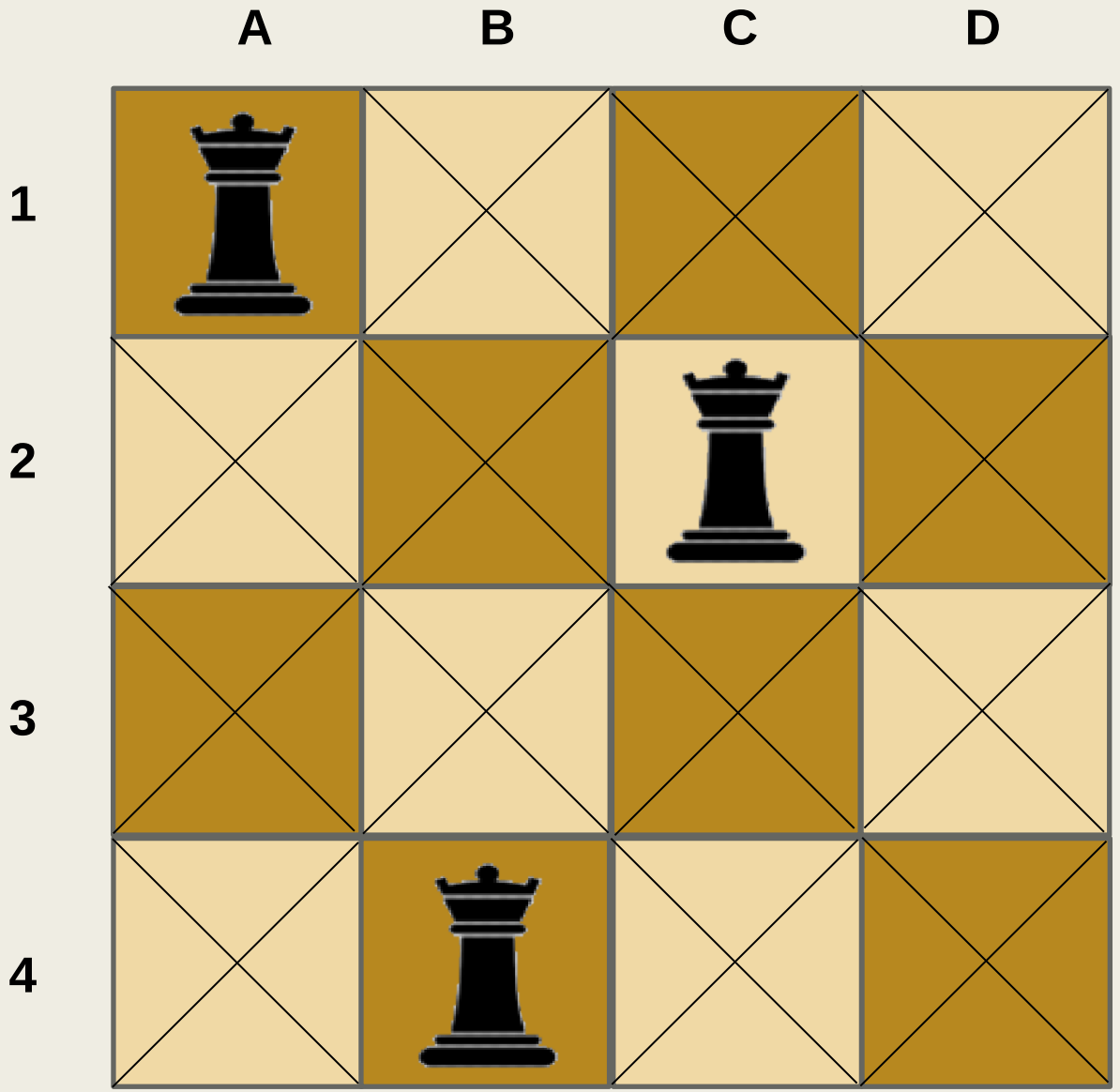
9663 N-Queen

N = 4



9663 N-Queen

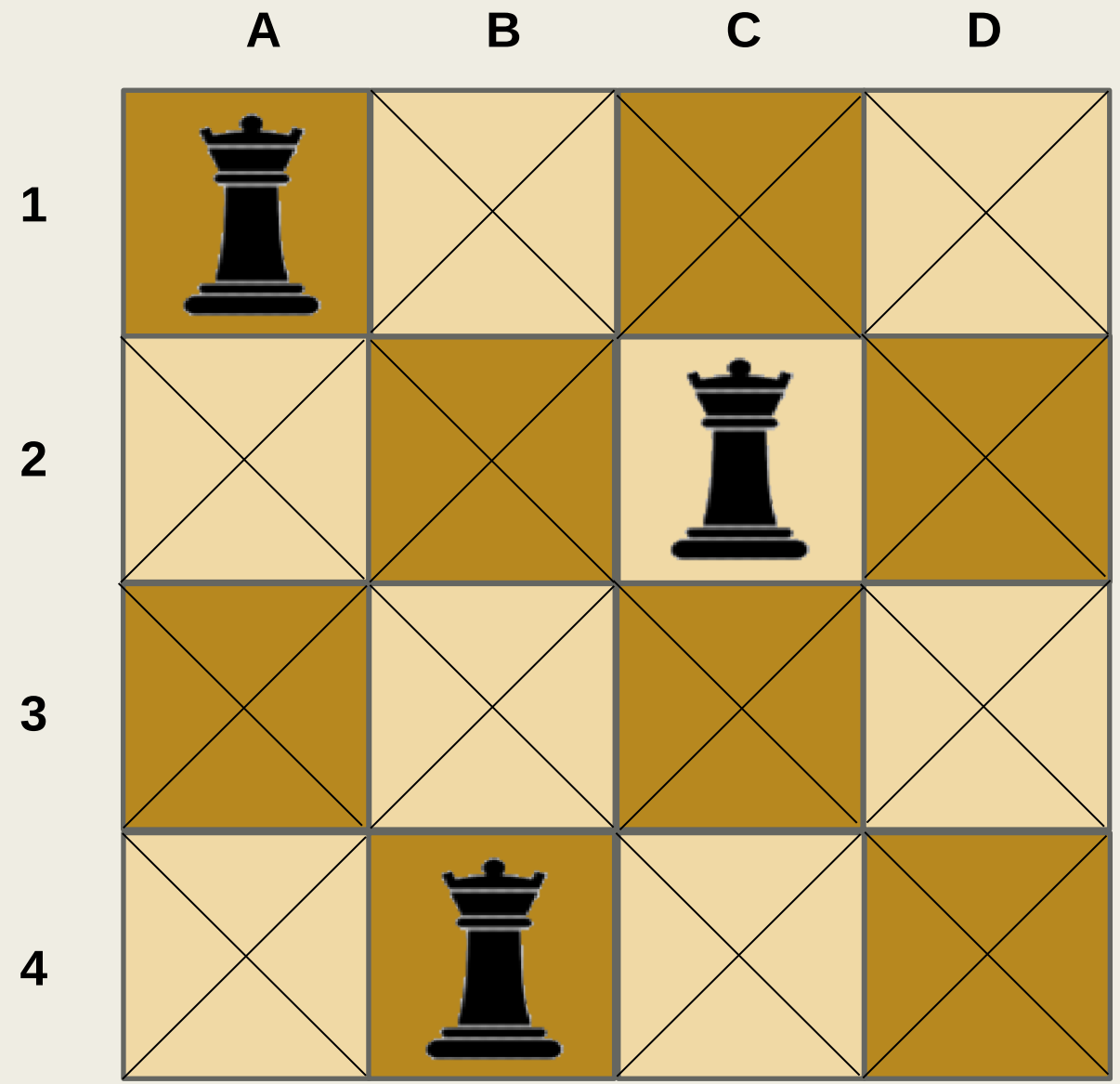
N = 4



9663 N-Queen

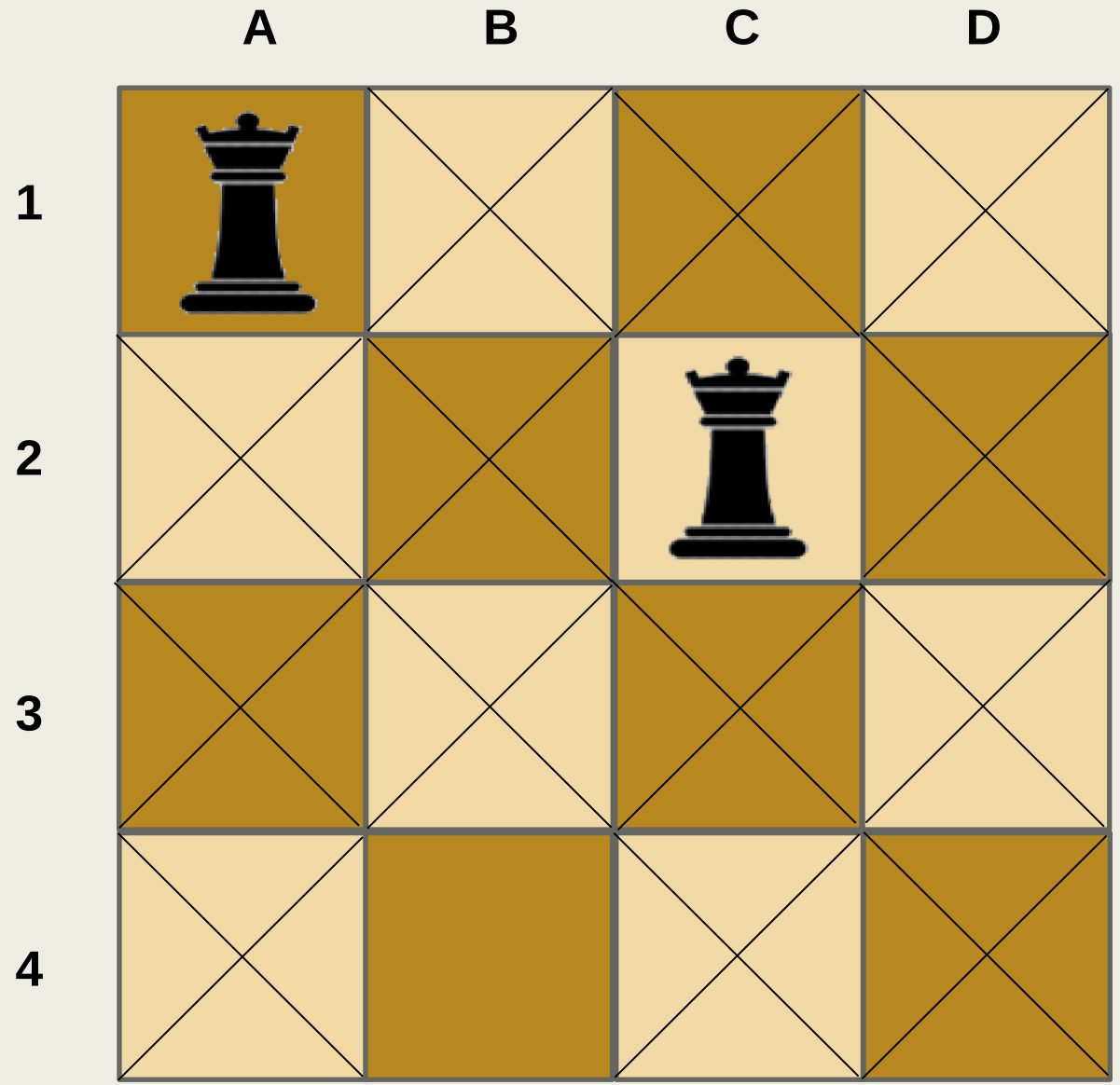
N = 4

4번째 퀸을 둘 자리가 없다!



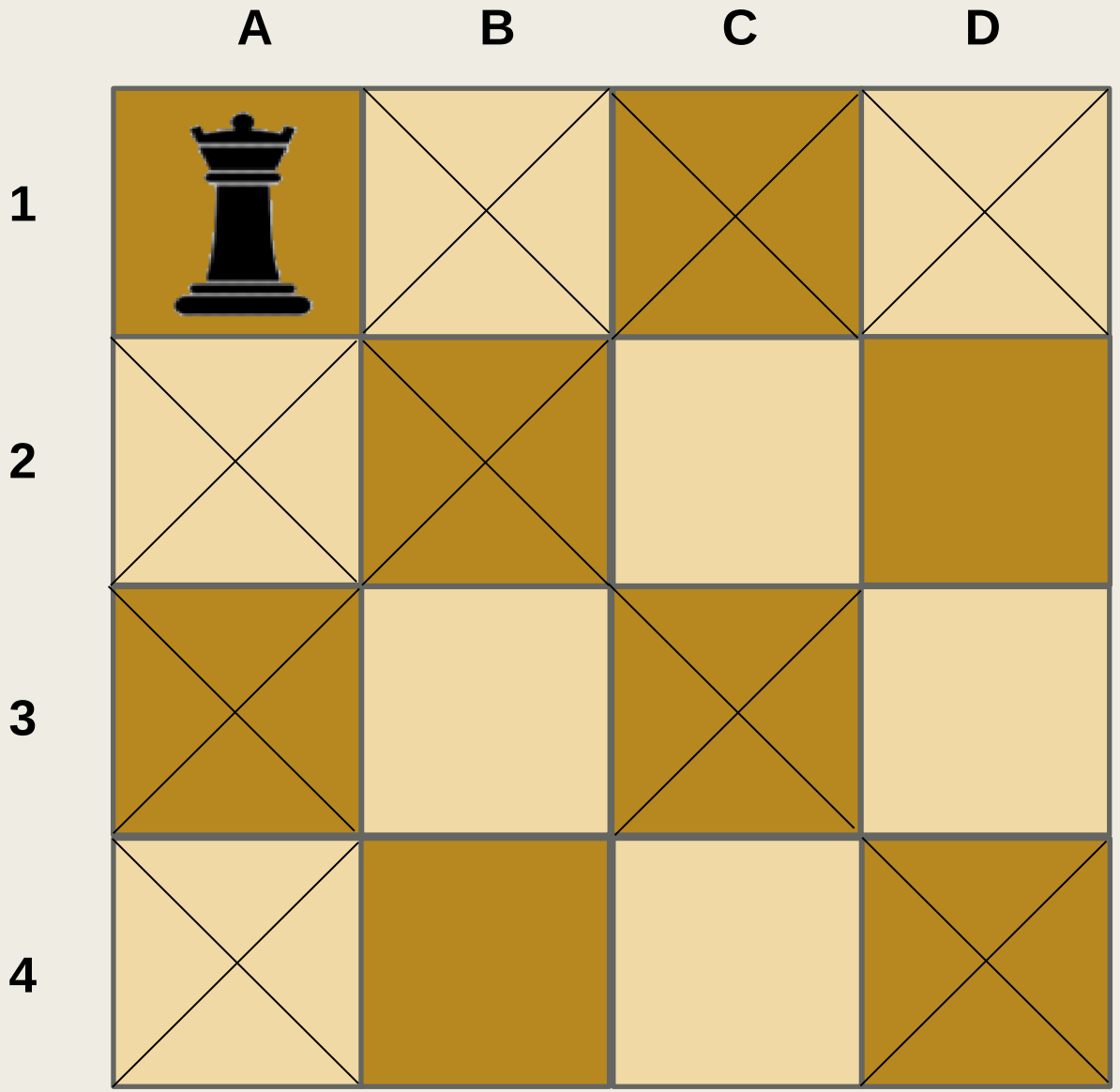
9663 N-Queen

N = 4



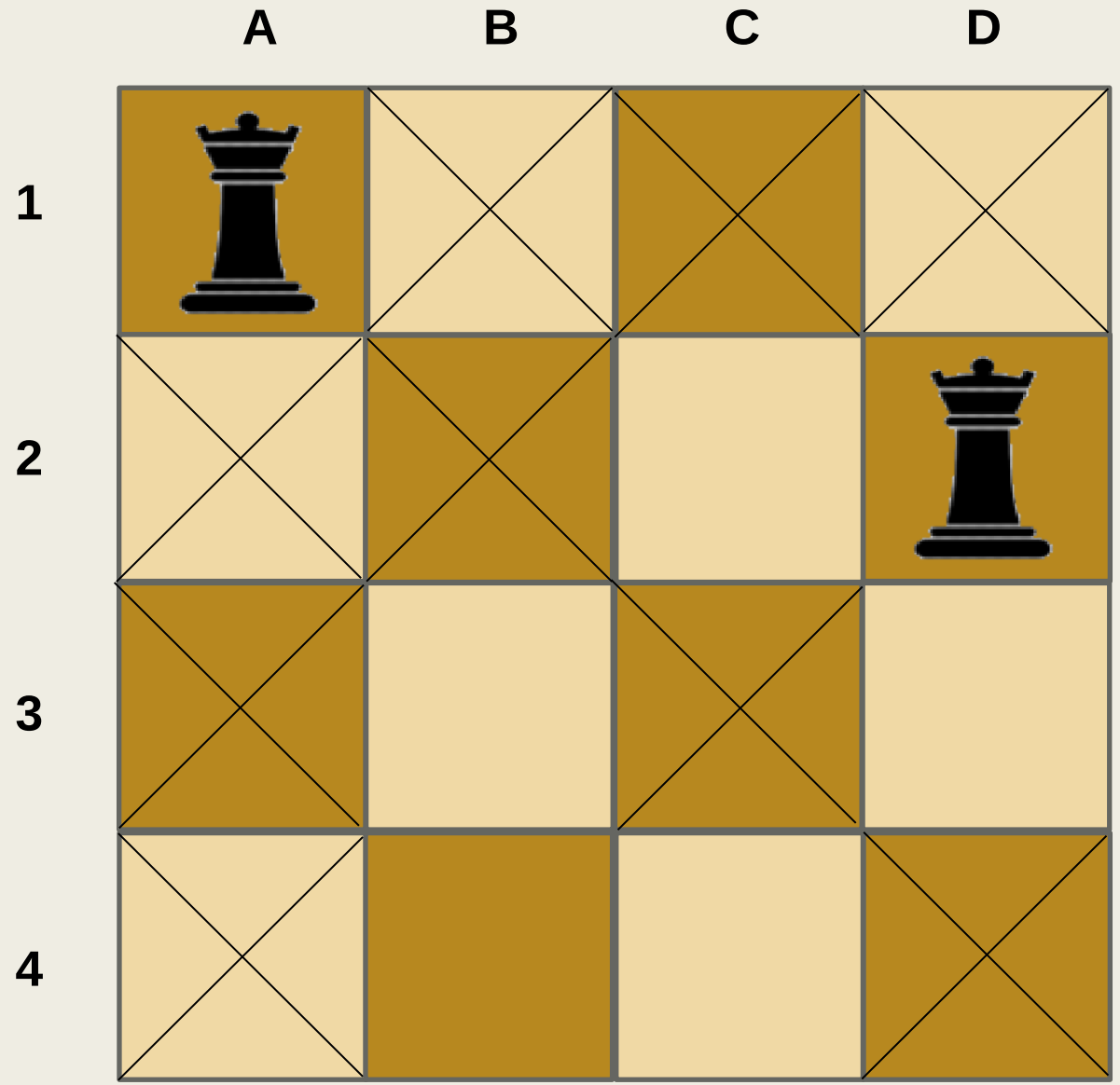
9663 N-Queen

N = 4



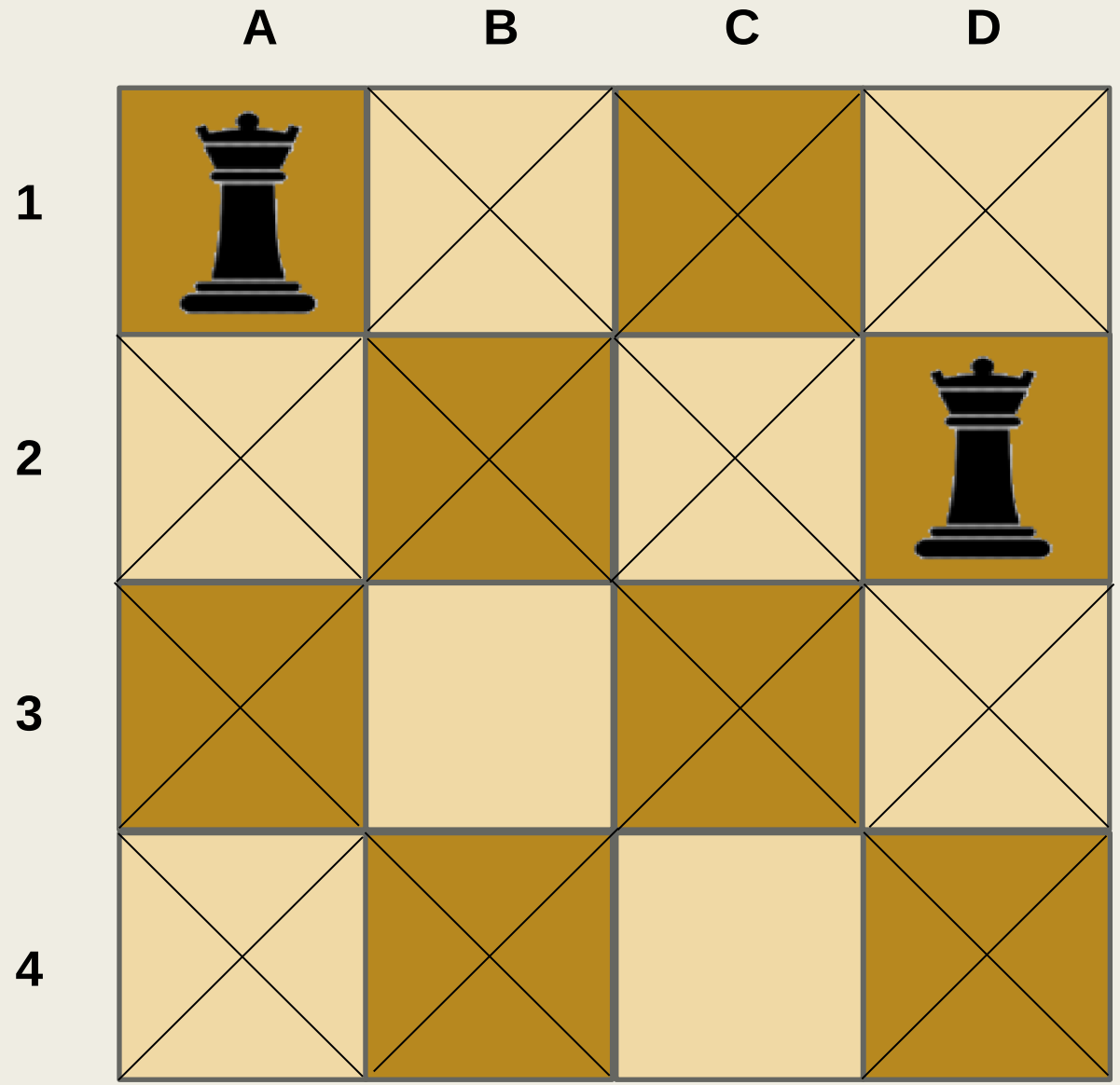
9663 N-Queen

N = 4



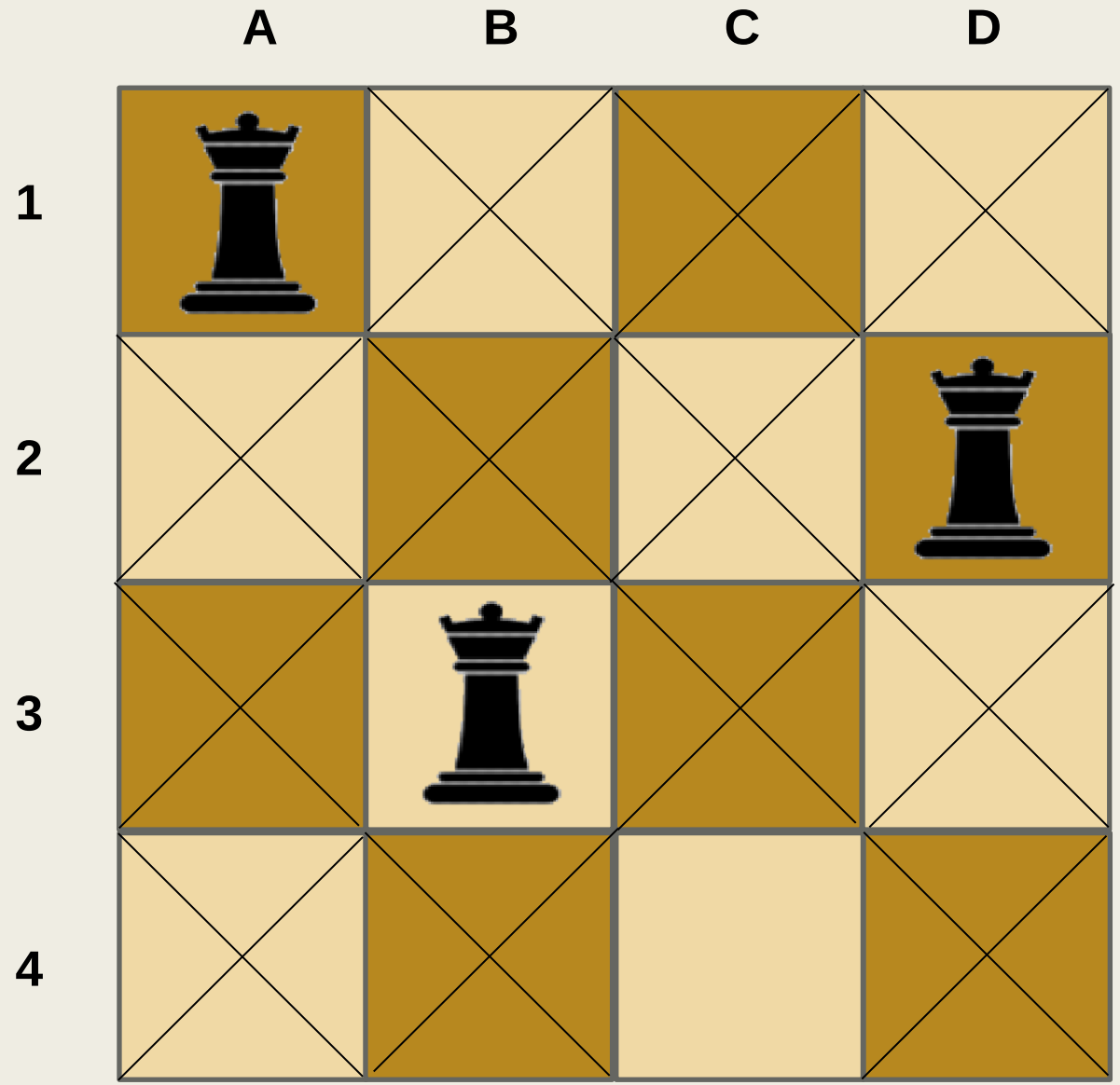
9663 N-Queen

N = 4



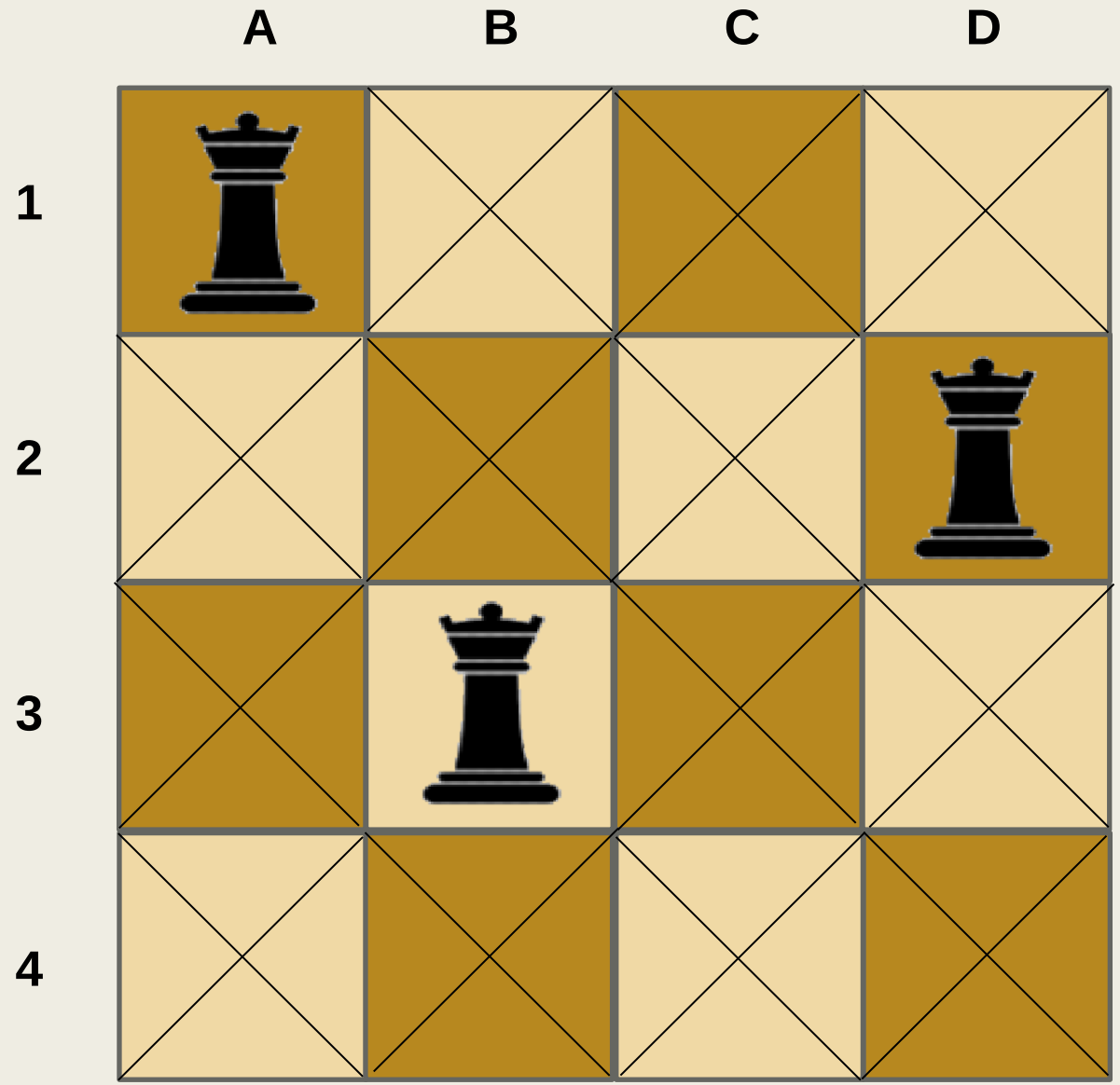
9663 N-Queen

N = 4



9663 N-Queen

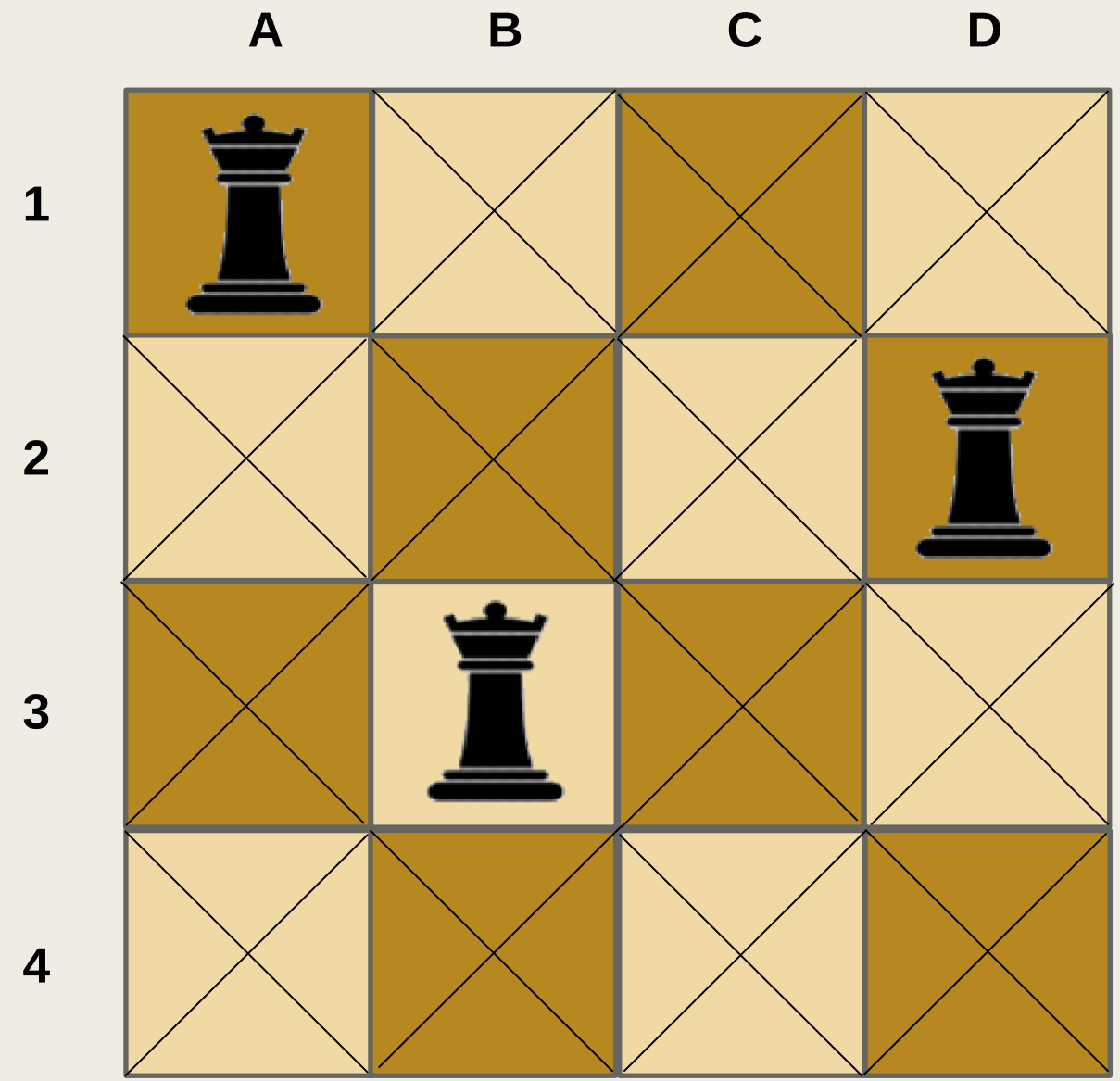
N = 4



9663 N-Queen

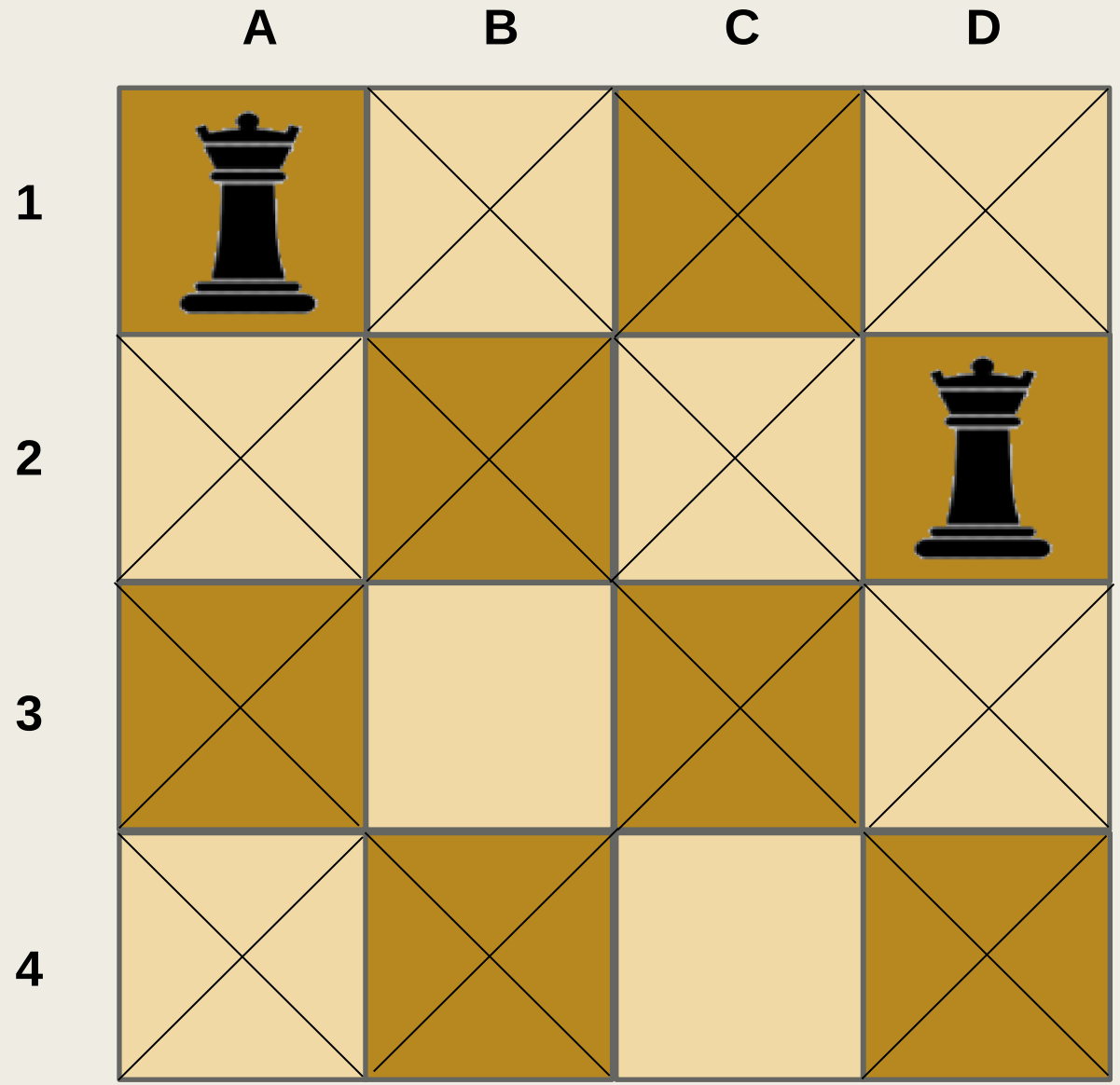
N = 4

4번째 퀸을 둘 자리가 없다!



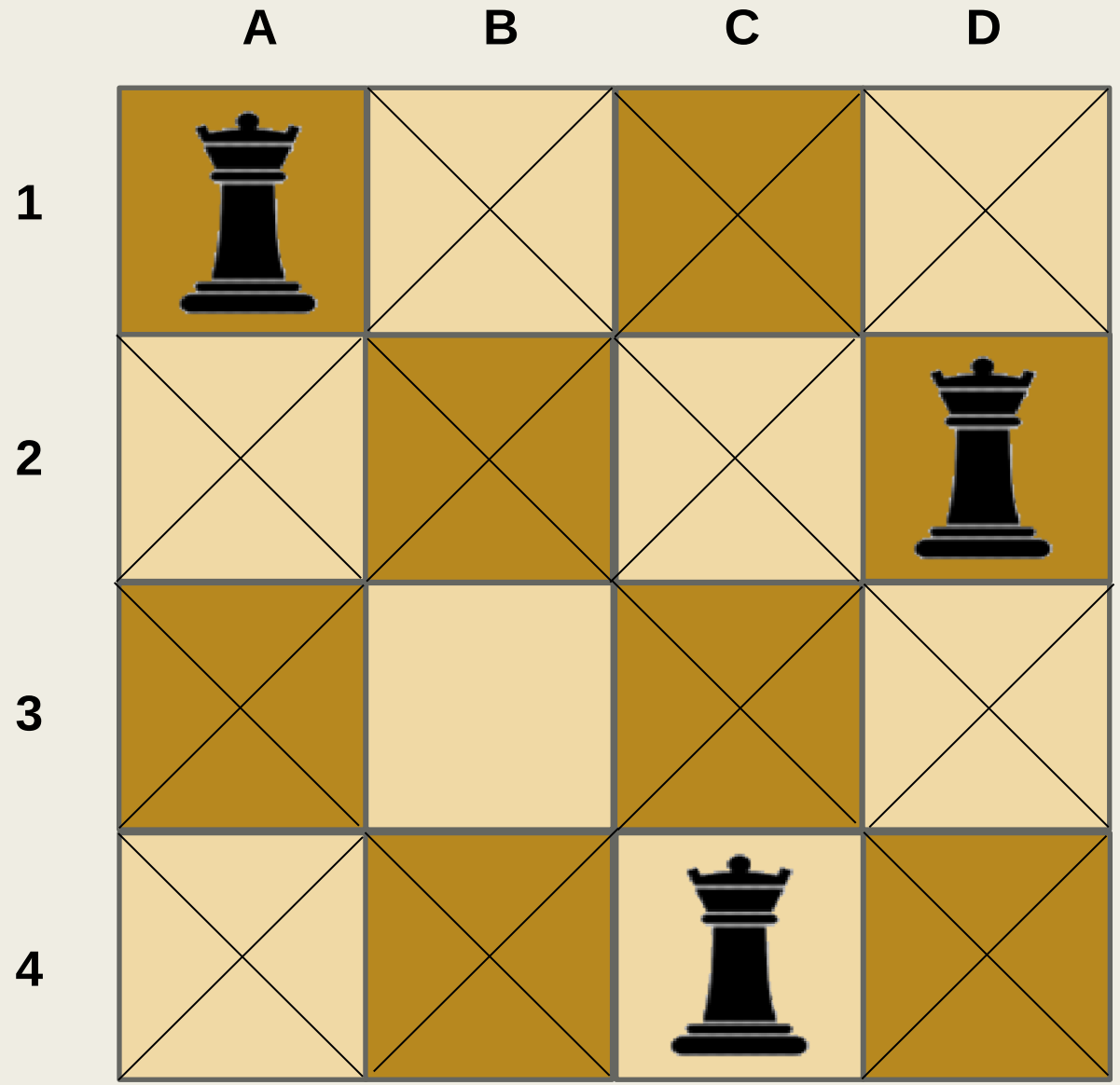
9663 N-Queen

N = 4



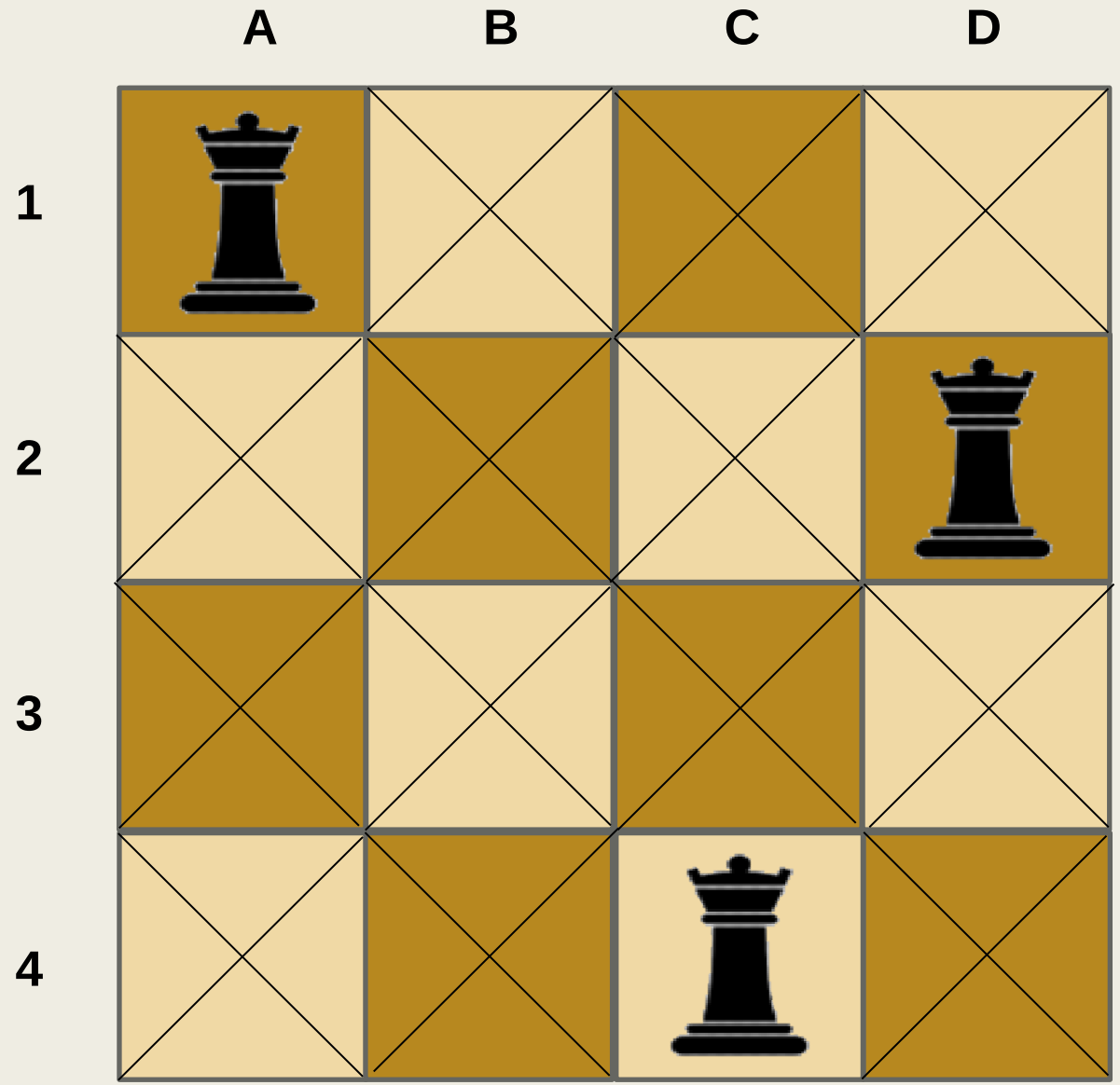
9663 N-Queen

N = 4



9663 N-Queen

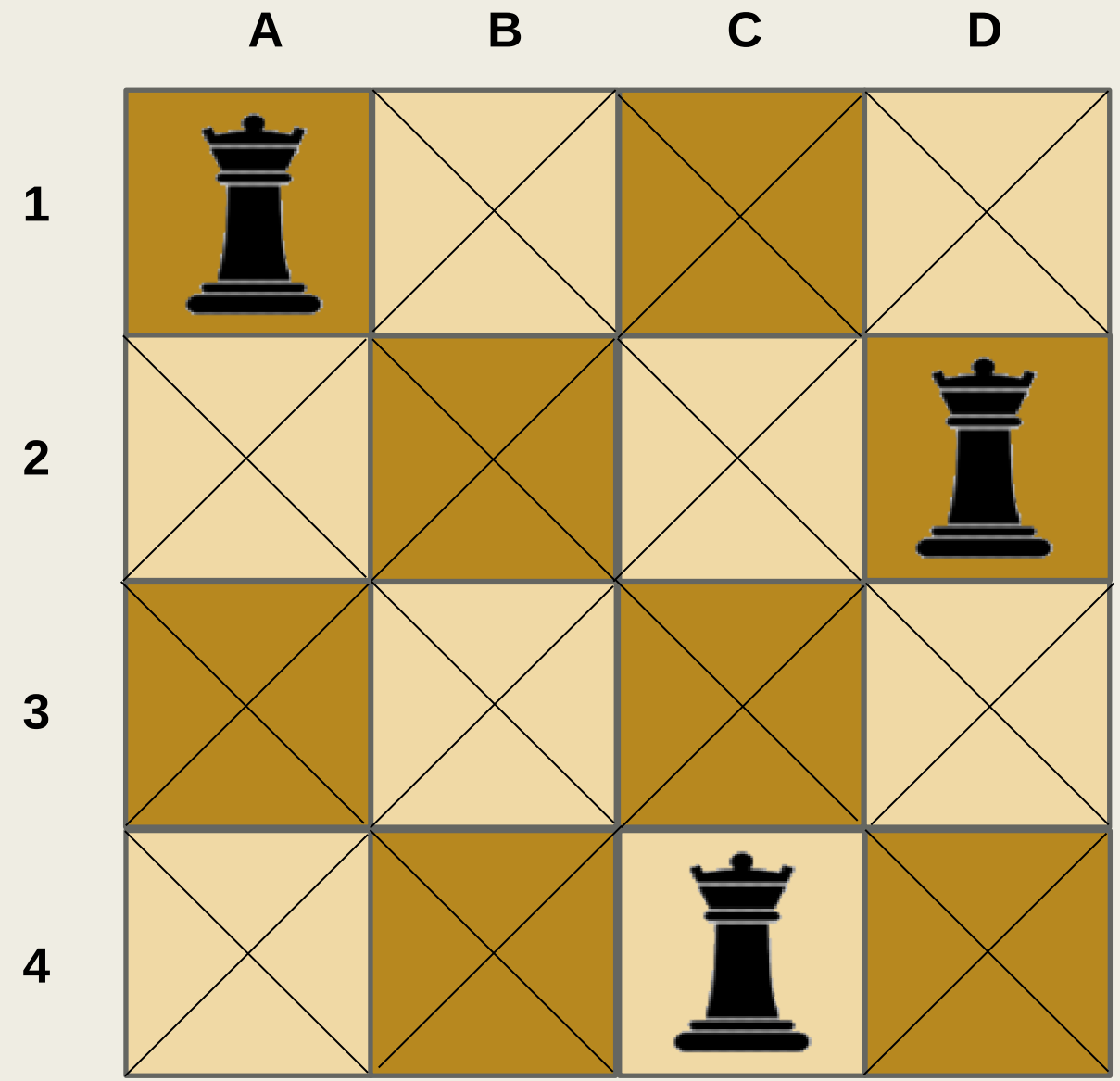
N = 4



9663 N-Queen

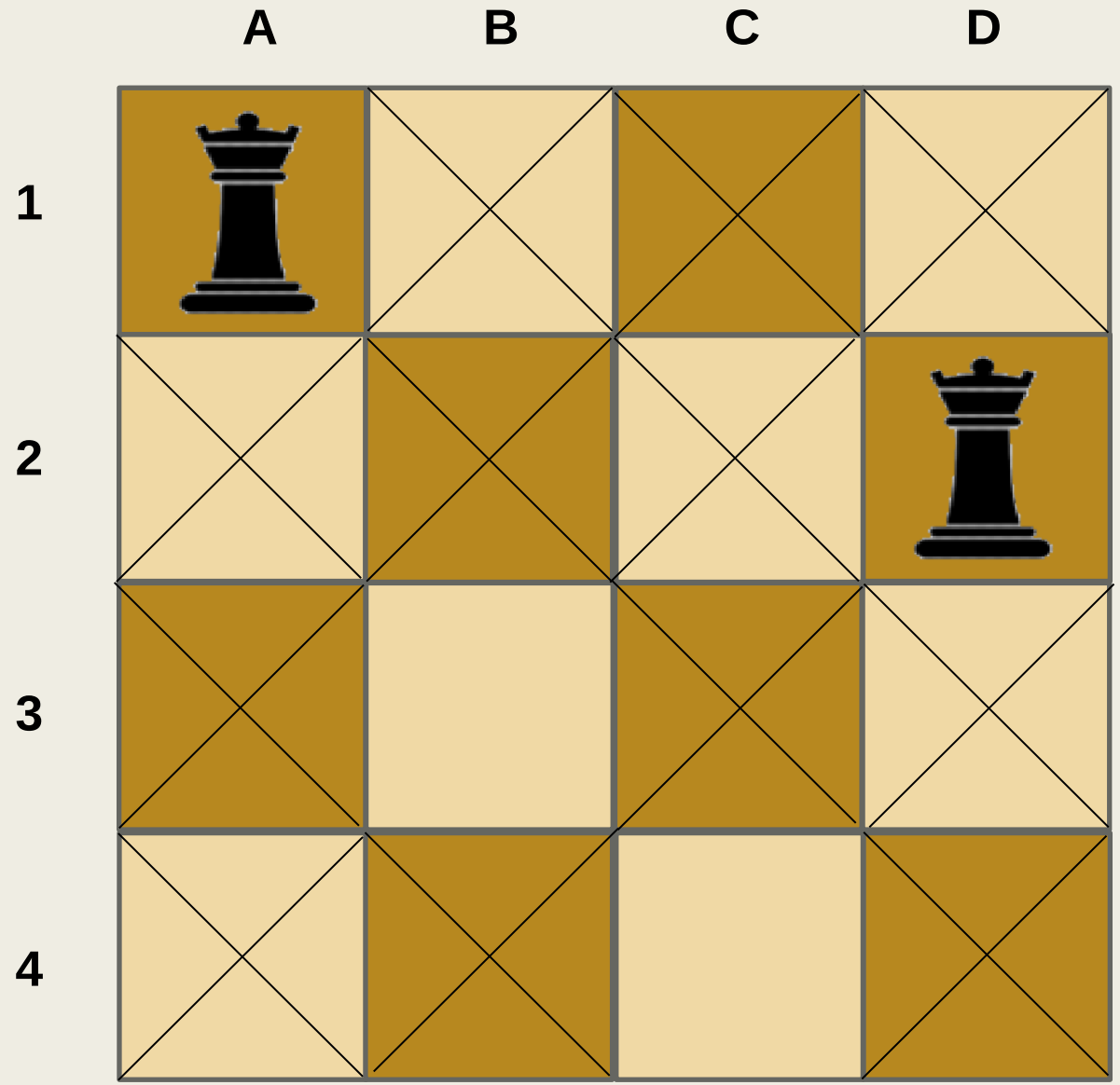
N = 4

4번째 퀸을 둘 자리가 없다!



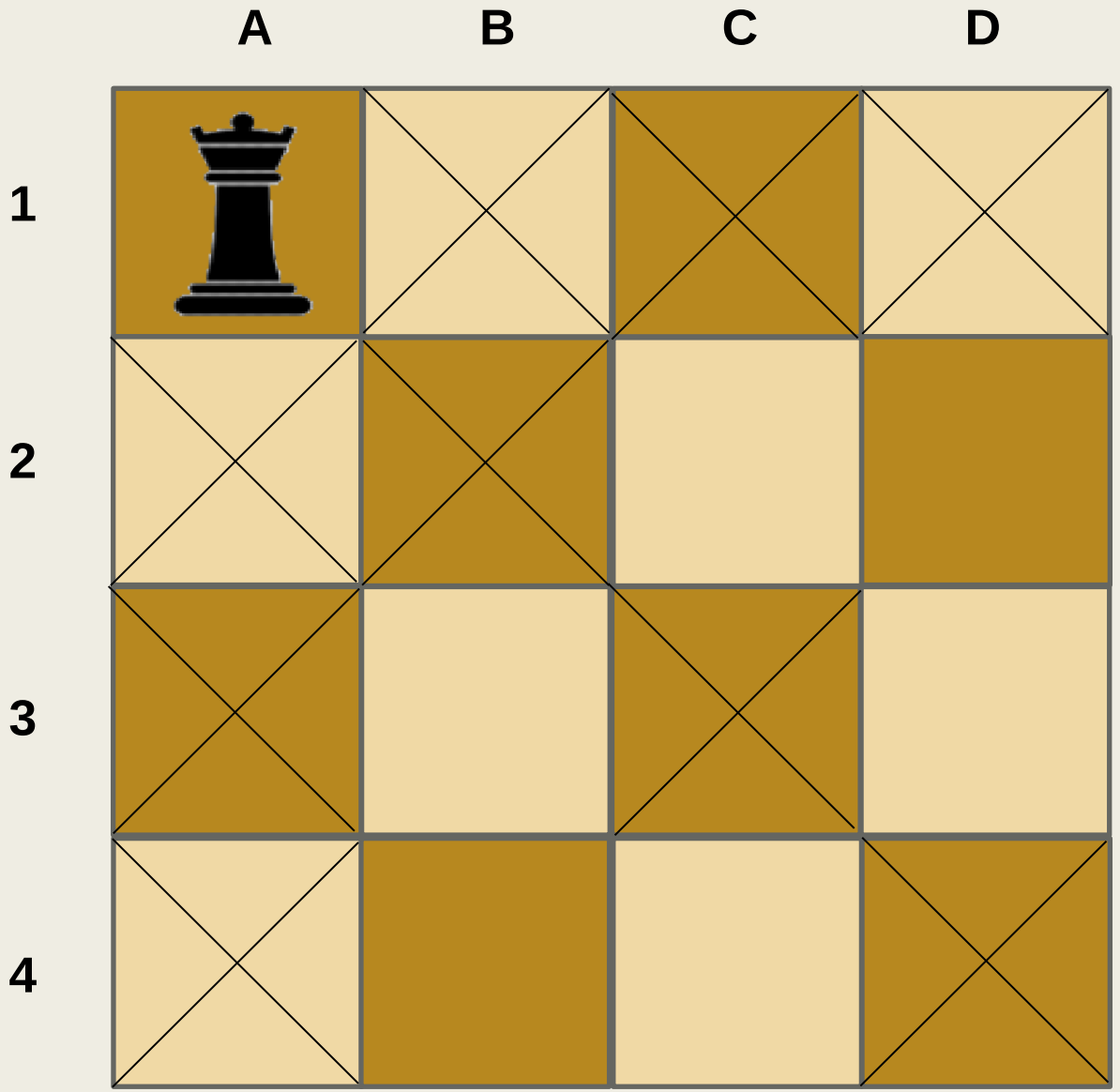
9663 N-Queen

N = 4



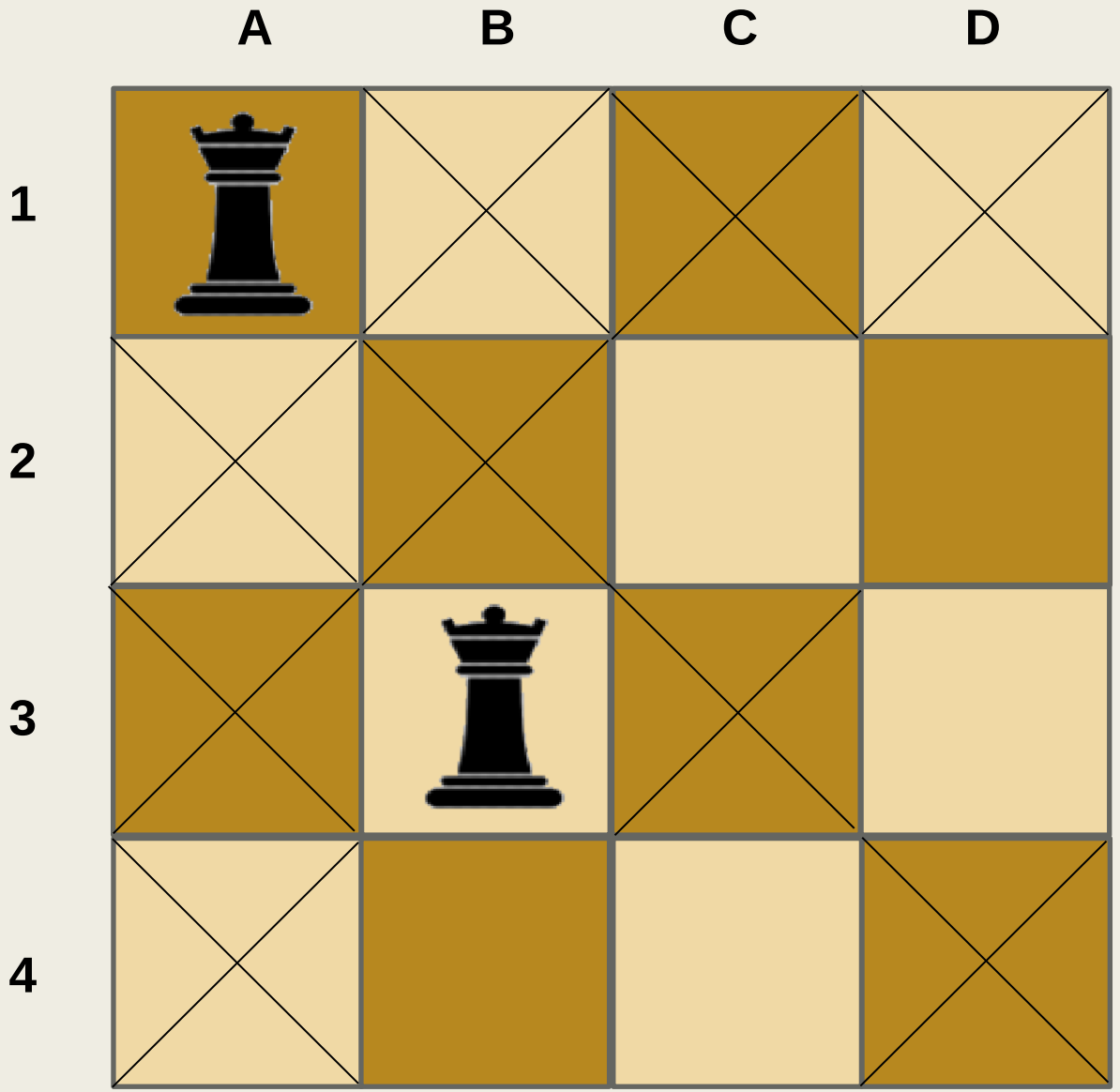
9663 N-Queen

N = 4



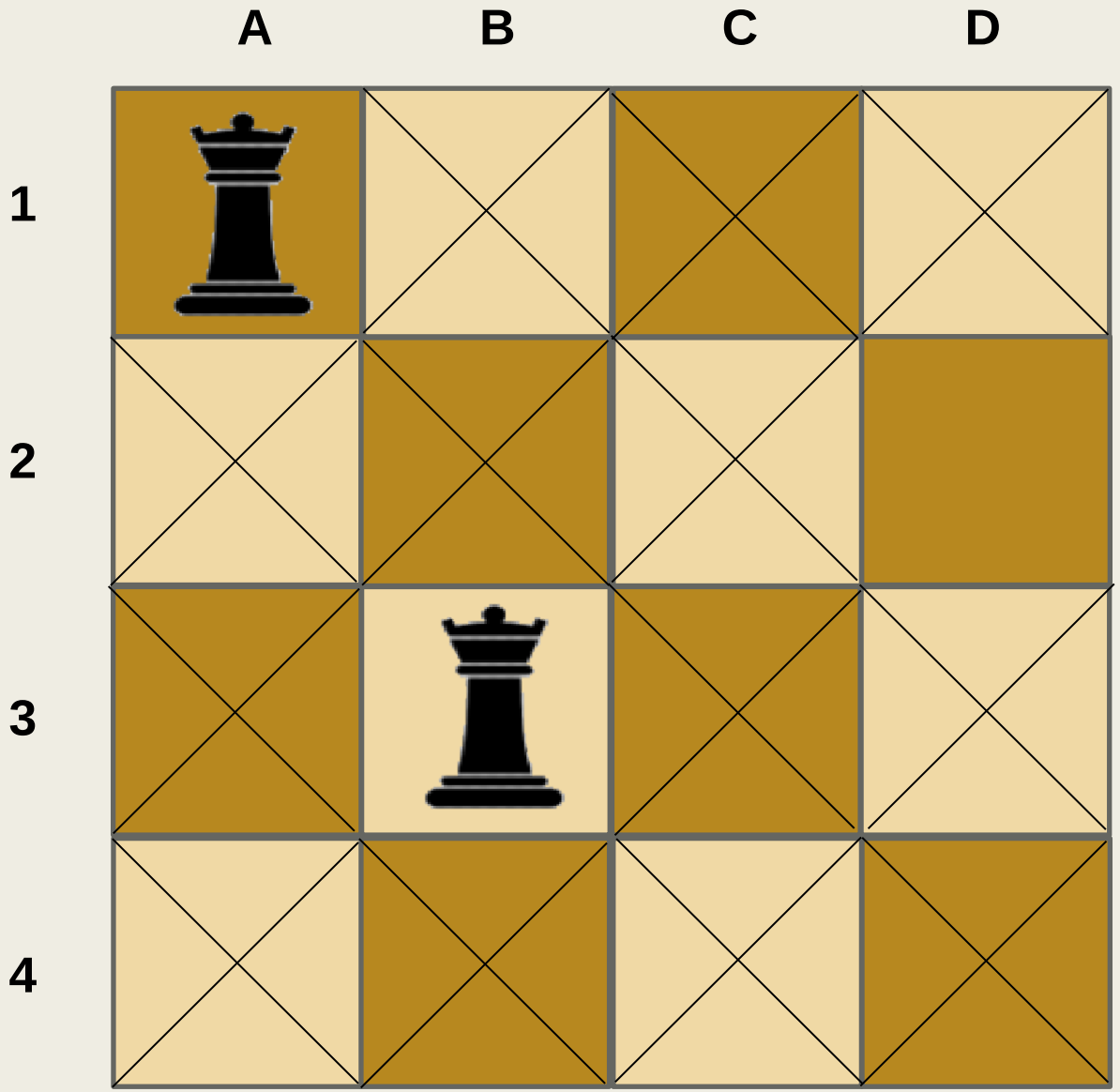
9663 N-Queen

N = 4



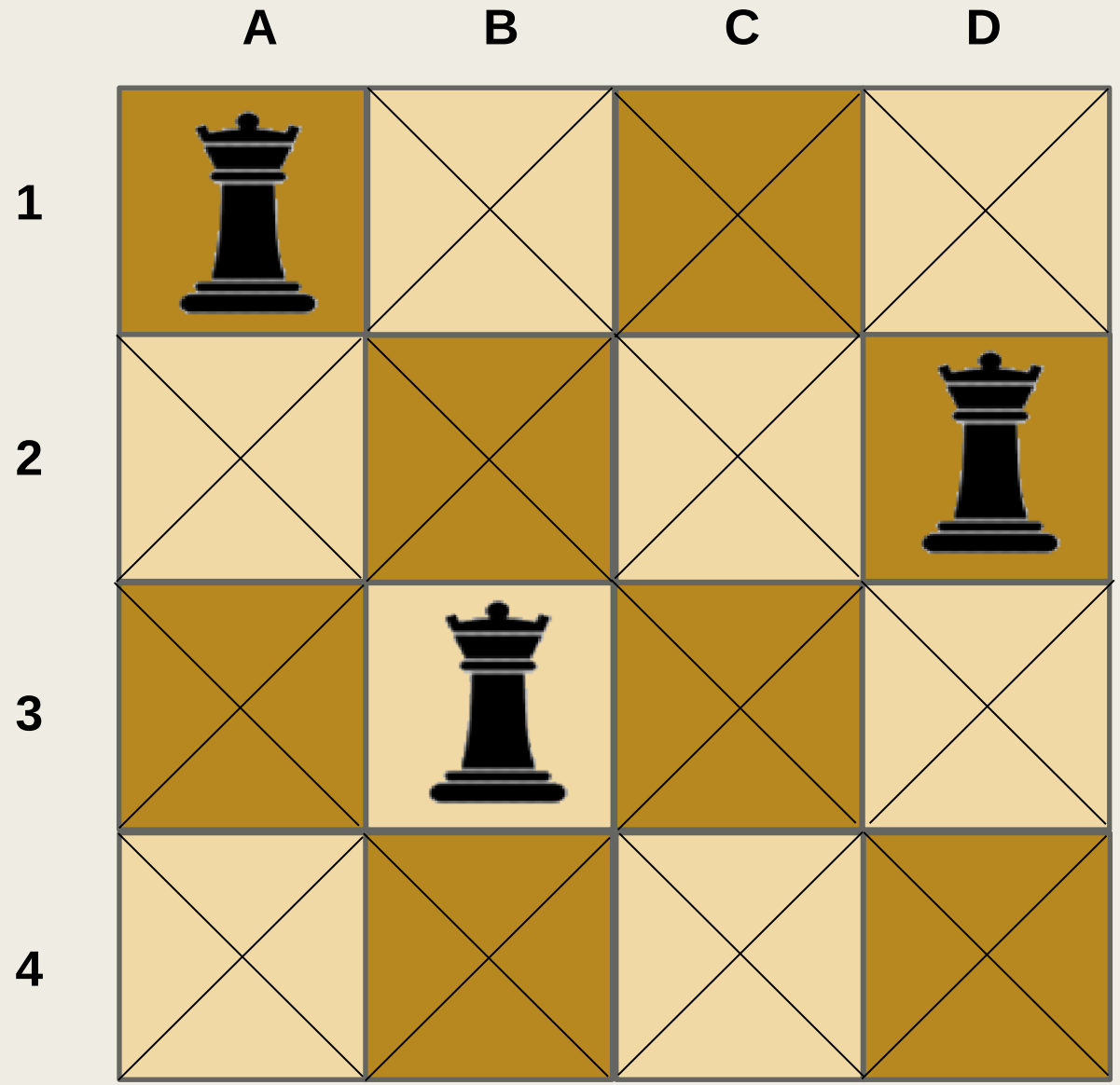
9663 N-Queen

N = 4



9663 N-Queen

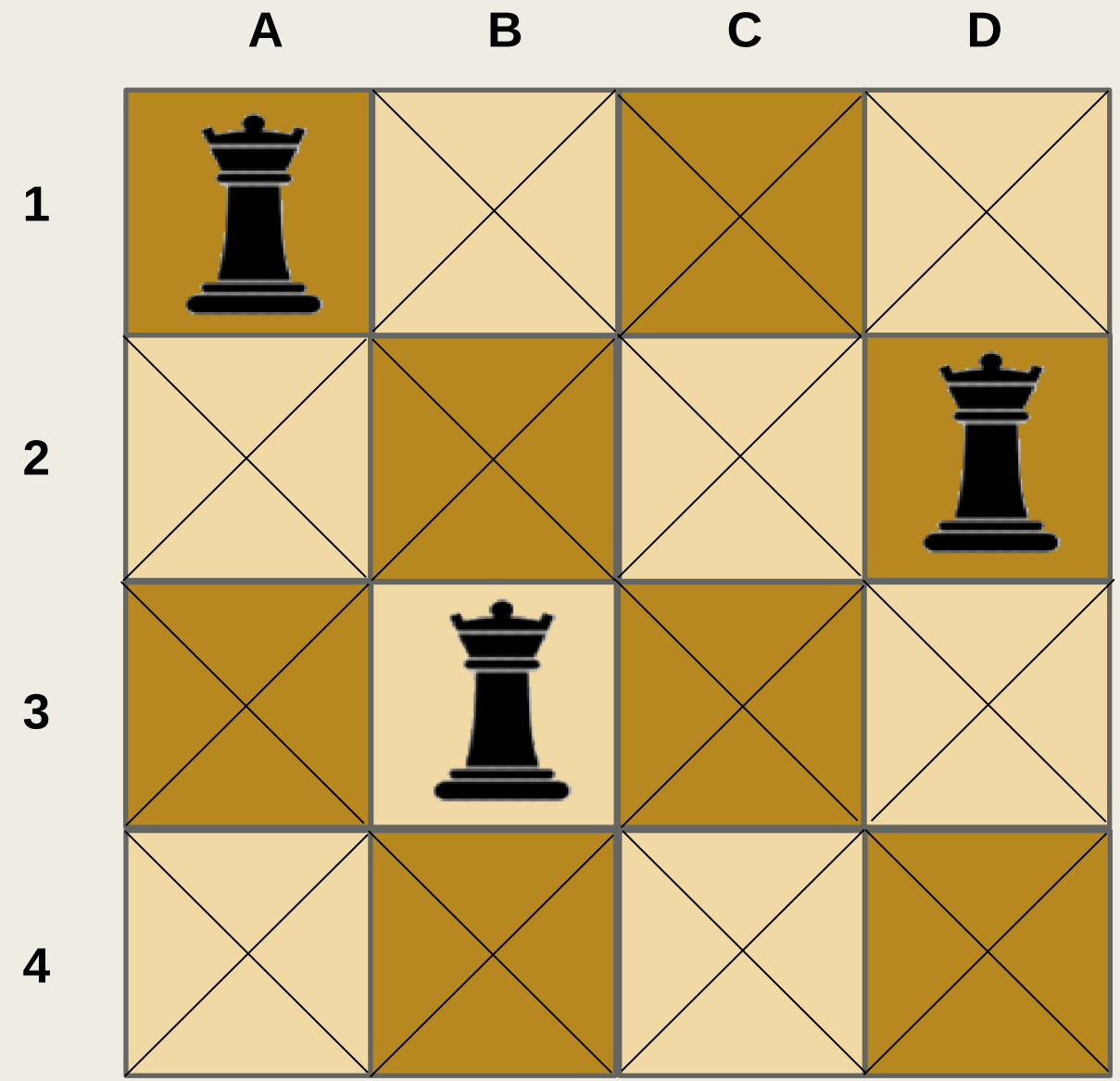
N = 4



9663 N-Queen

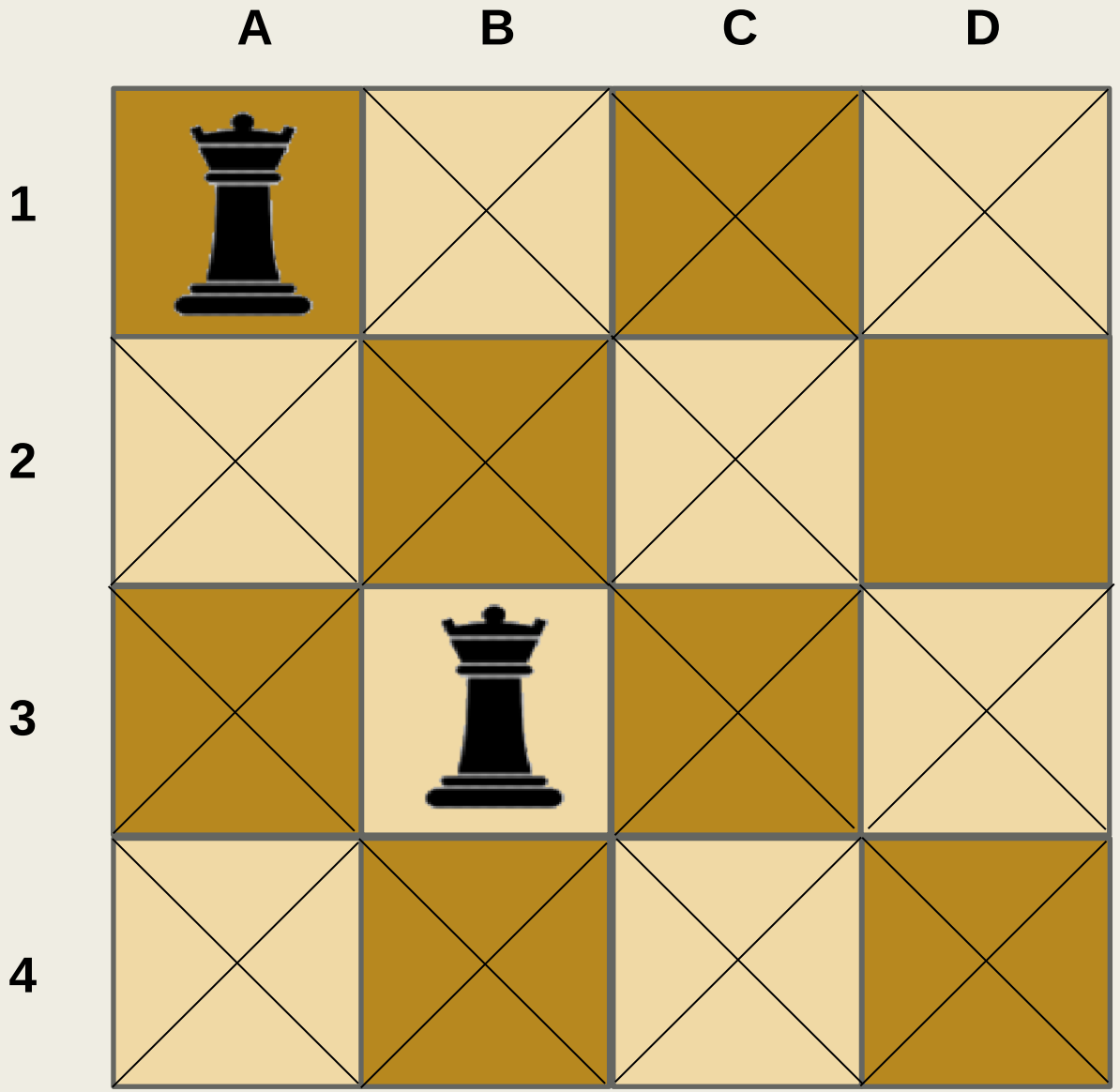
N = 4

4번째 퀸을 둘 자리가 없다!



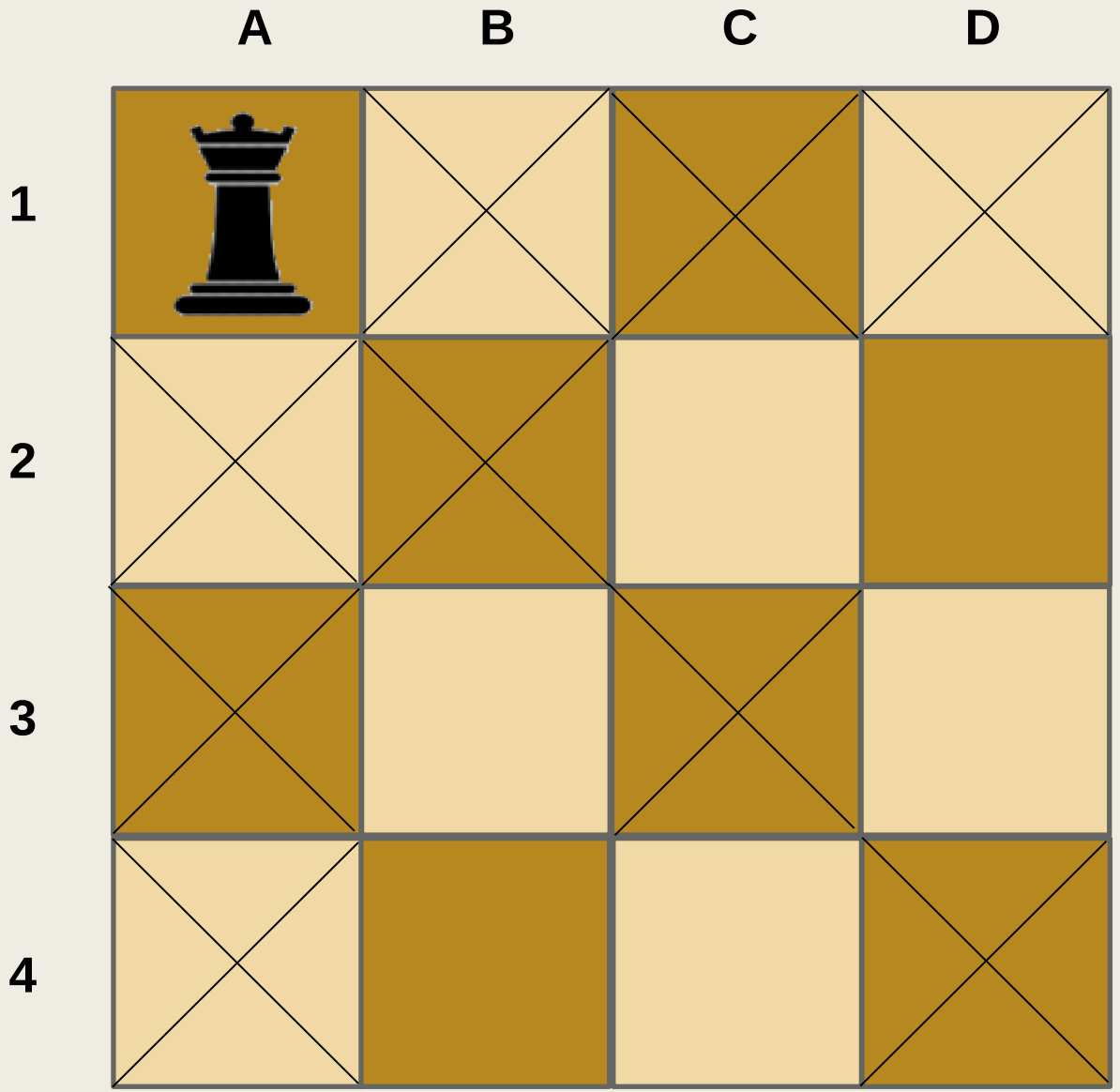
9663 N-Queen

N = 4



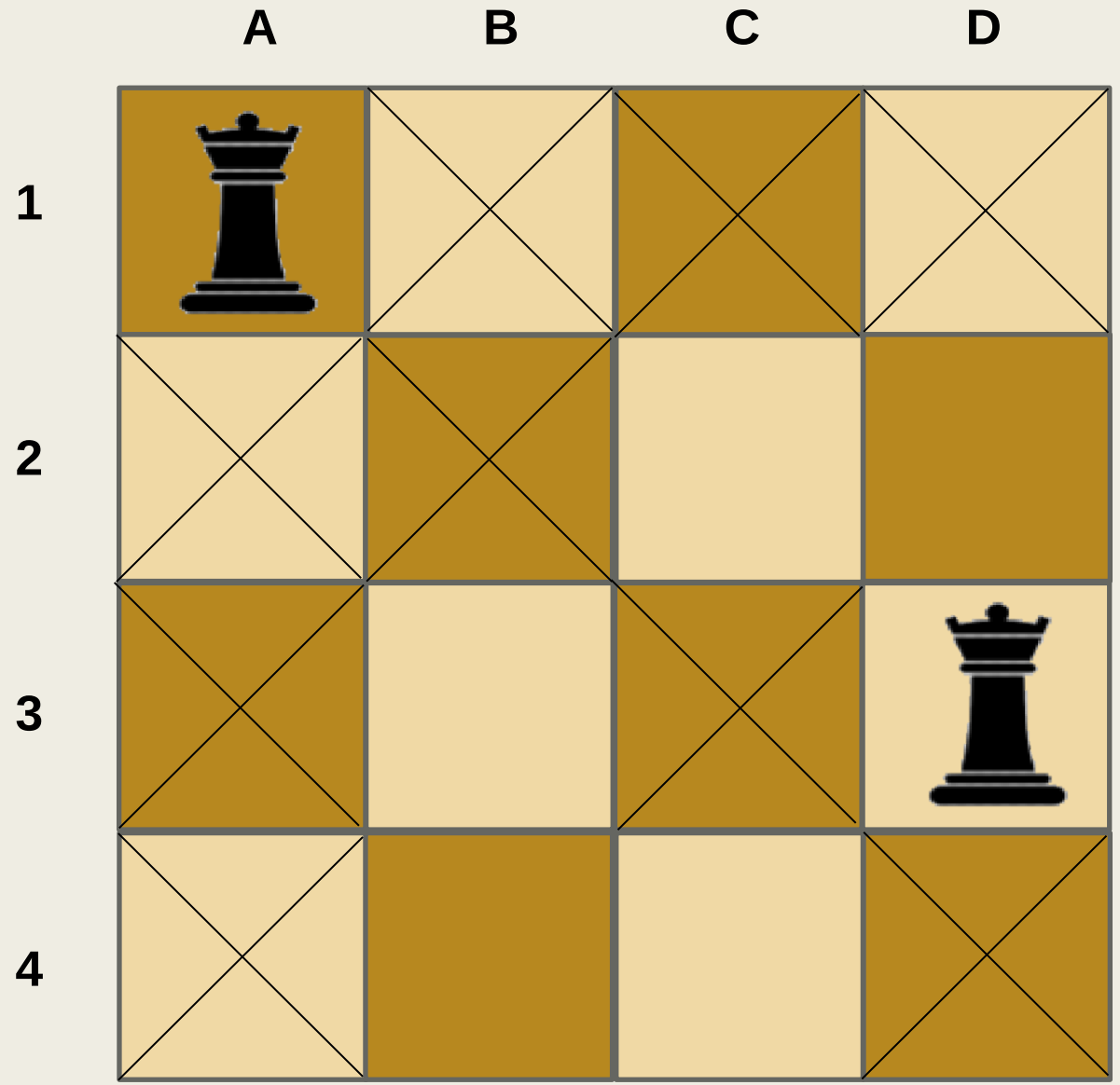
9663 N-Queen

N = 4



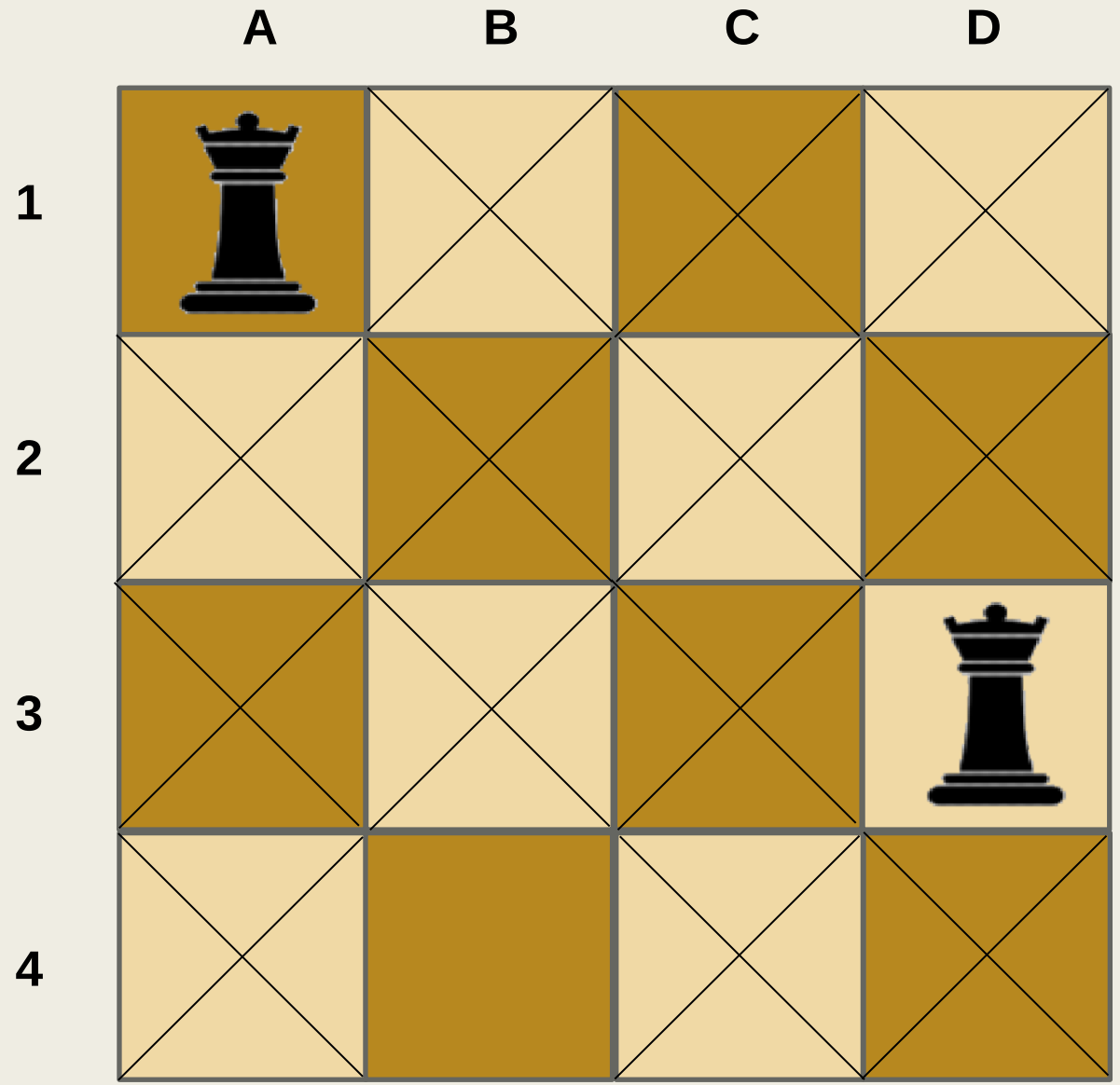
9663 N-Queen

N = 4



9663 N-Queen

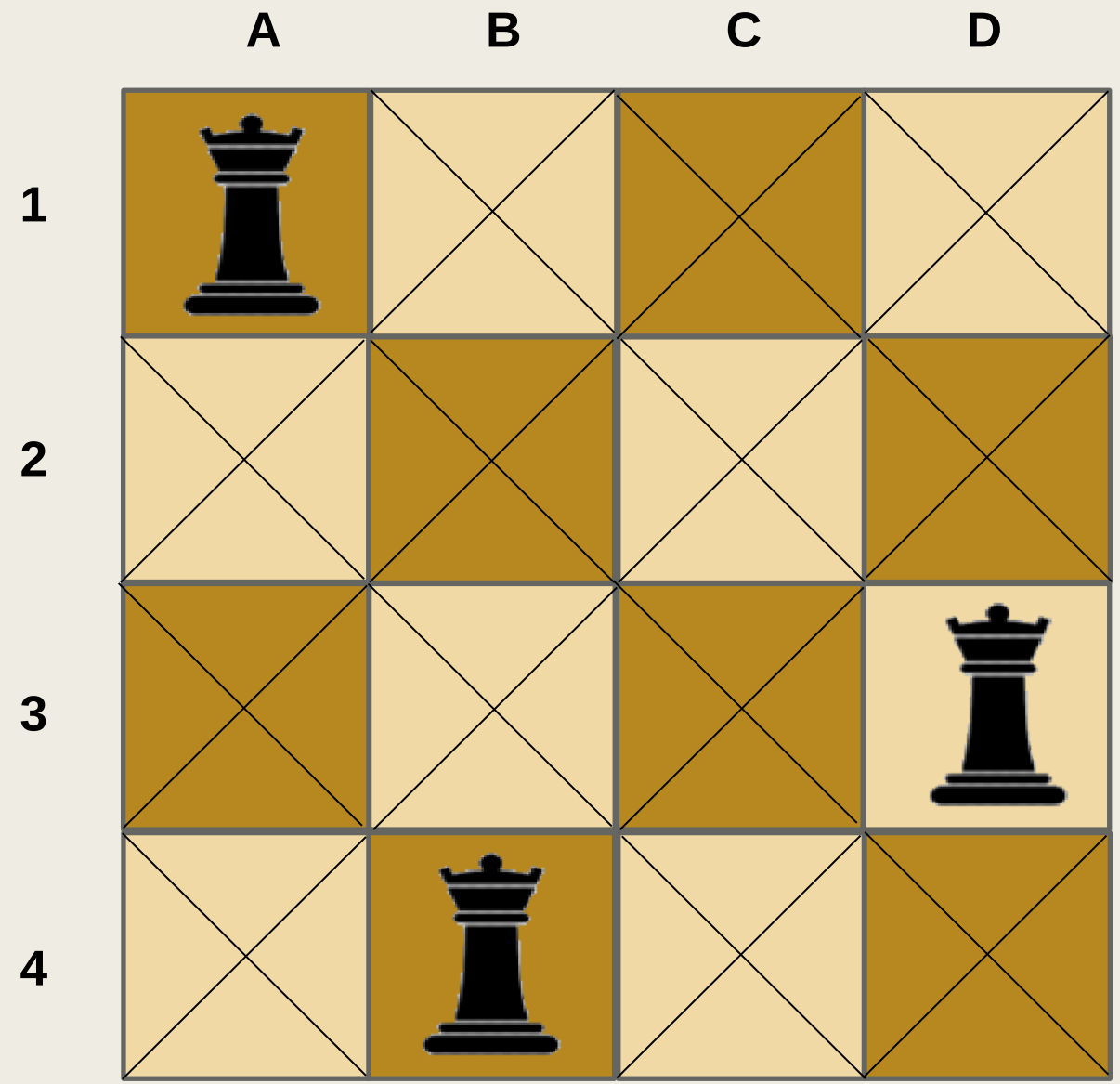
N = 4



9663 N-Queen

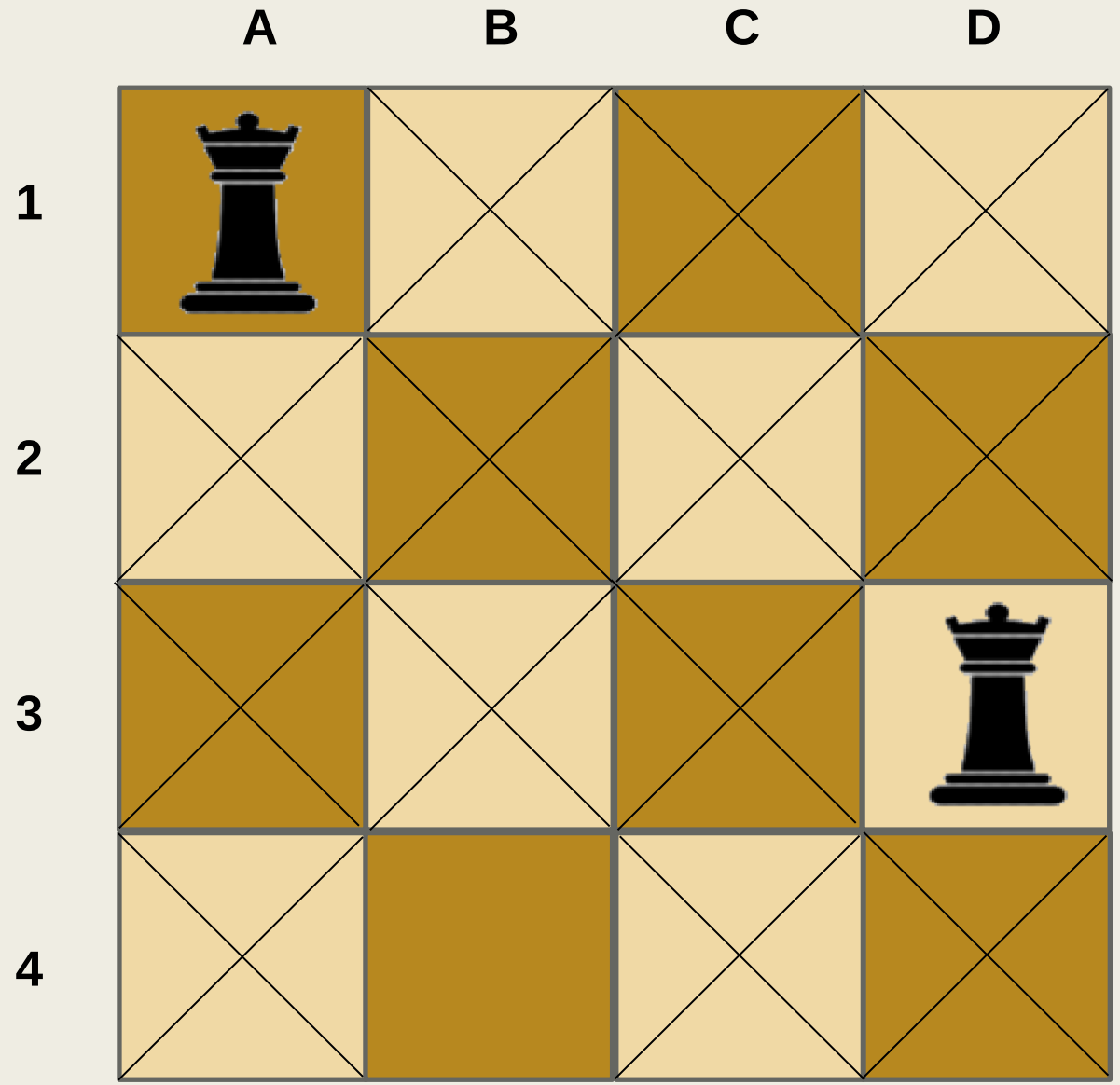
N = 4

4번째 퀸을 둘 자리가 없다!



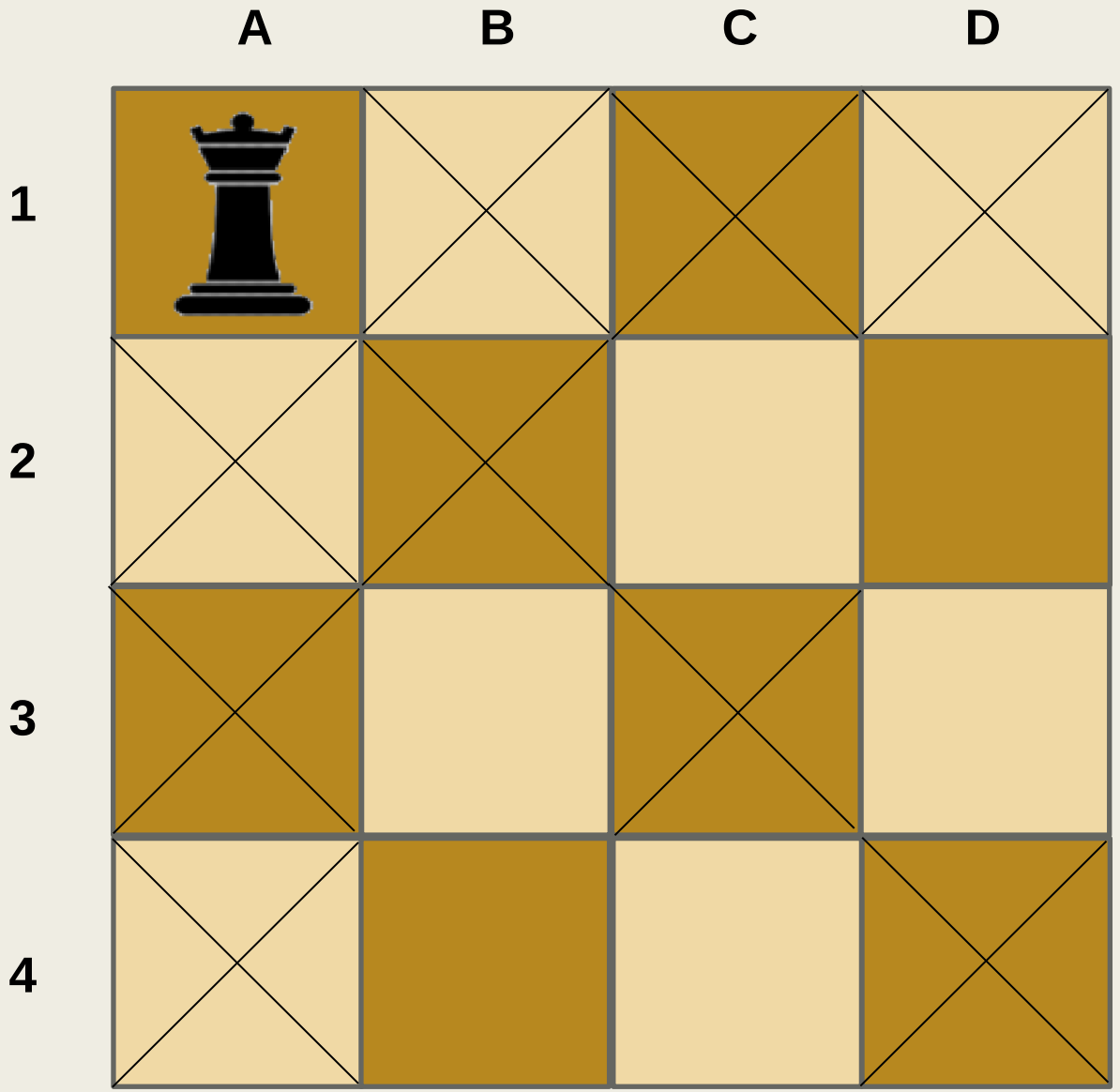
9663 N-Queen

N = 4



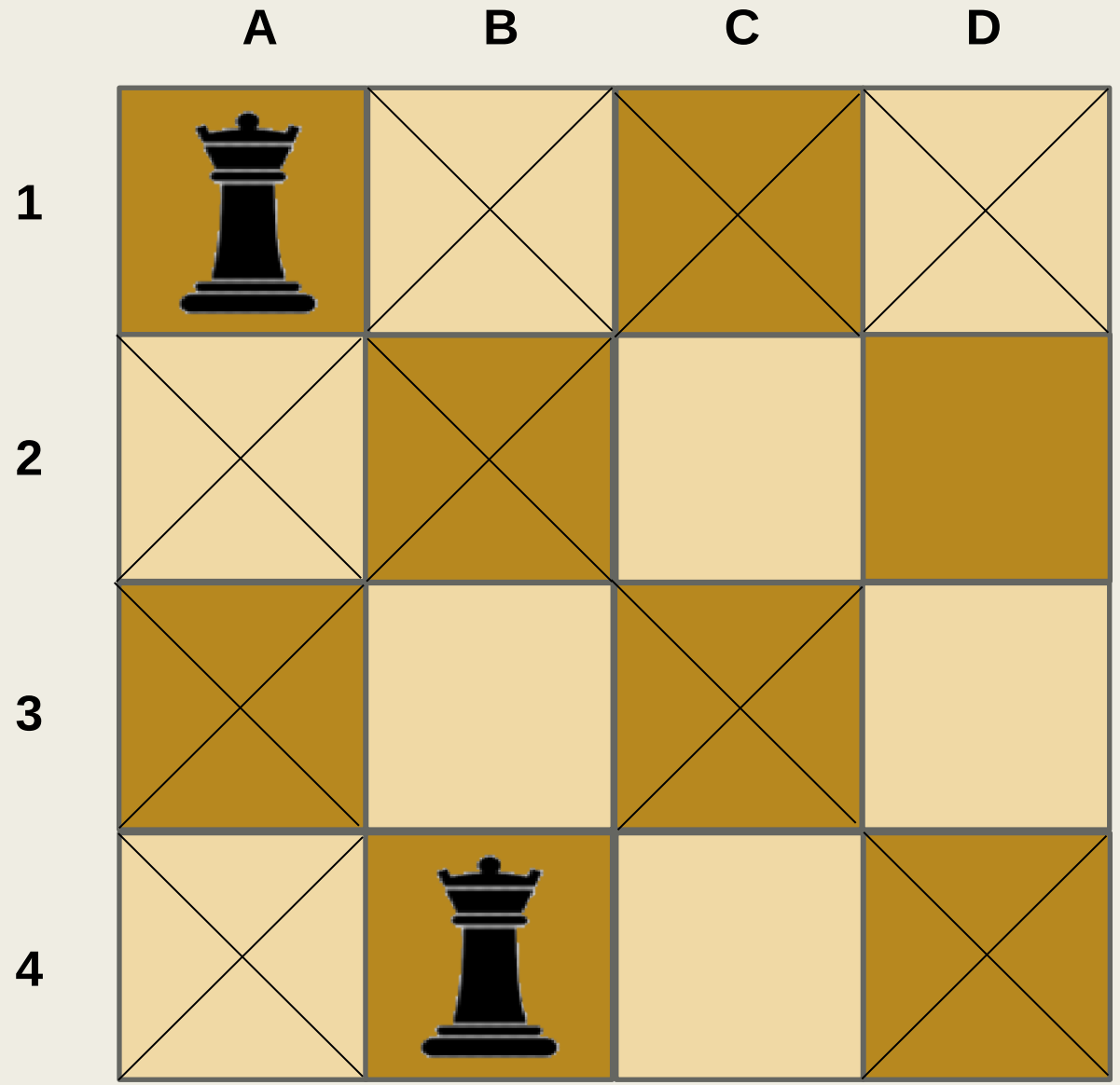
9663 N-Queen

N = 4



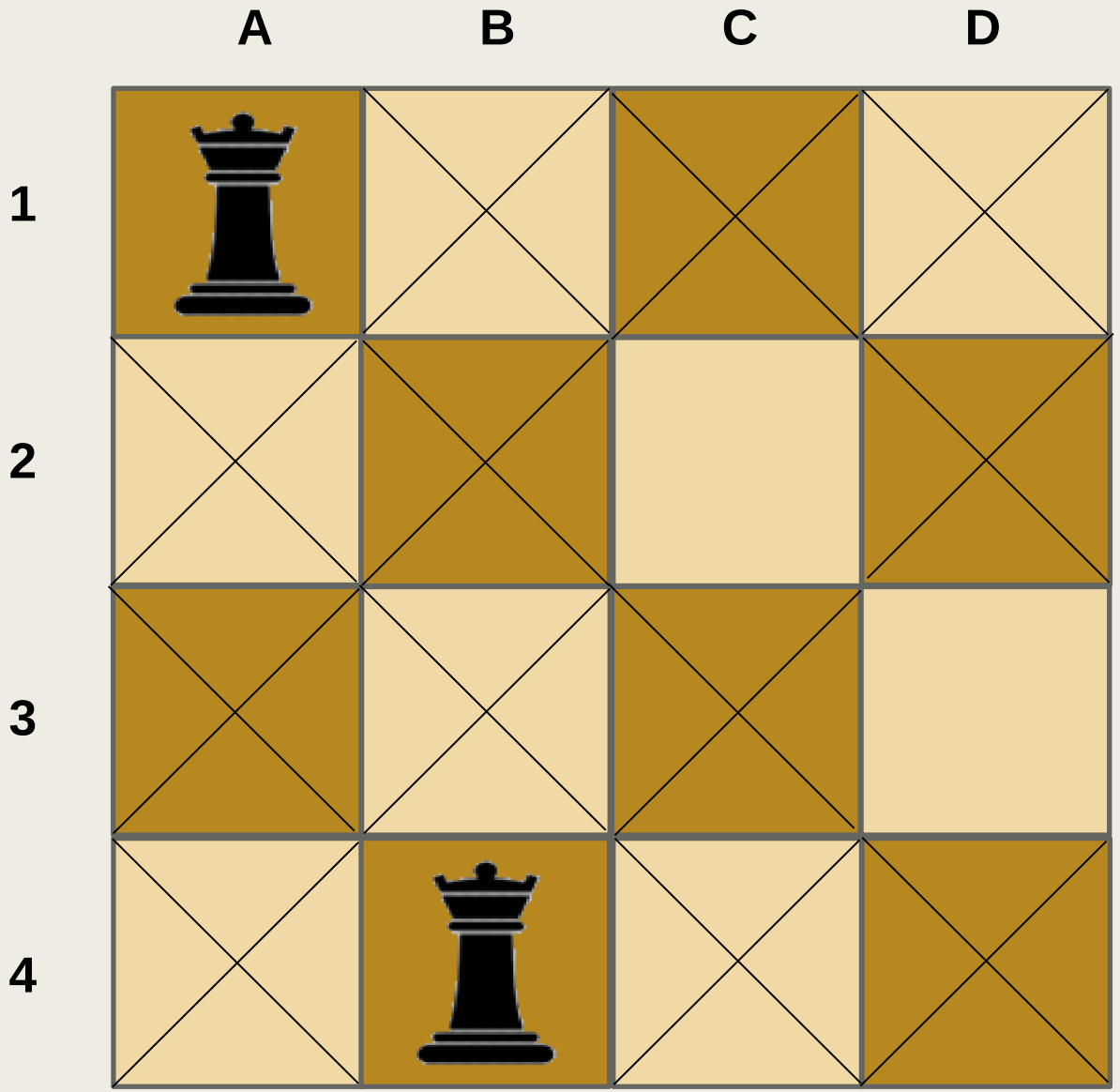
9663 N-Queen

N = 4



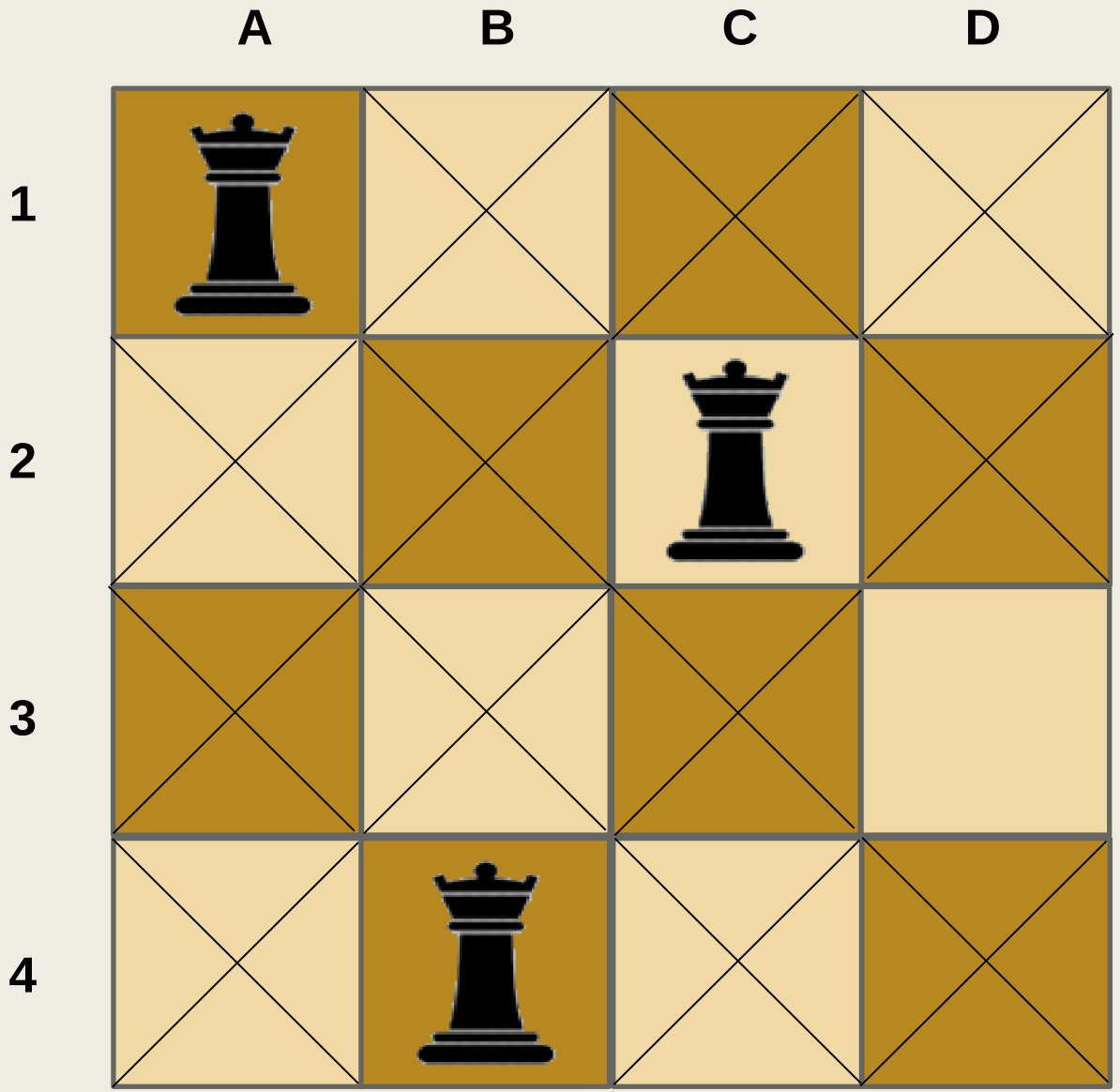
9663 N-Queen

N = 4



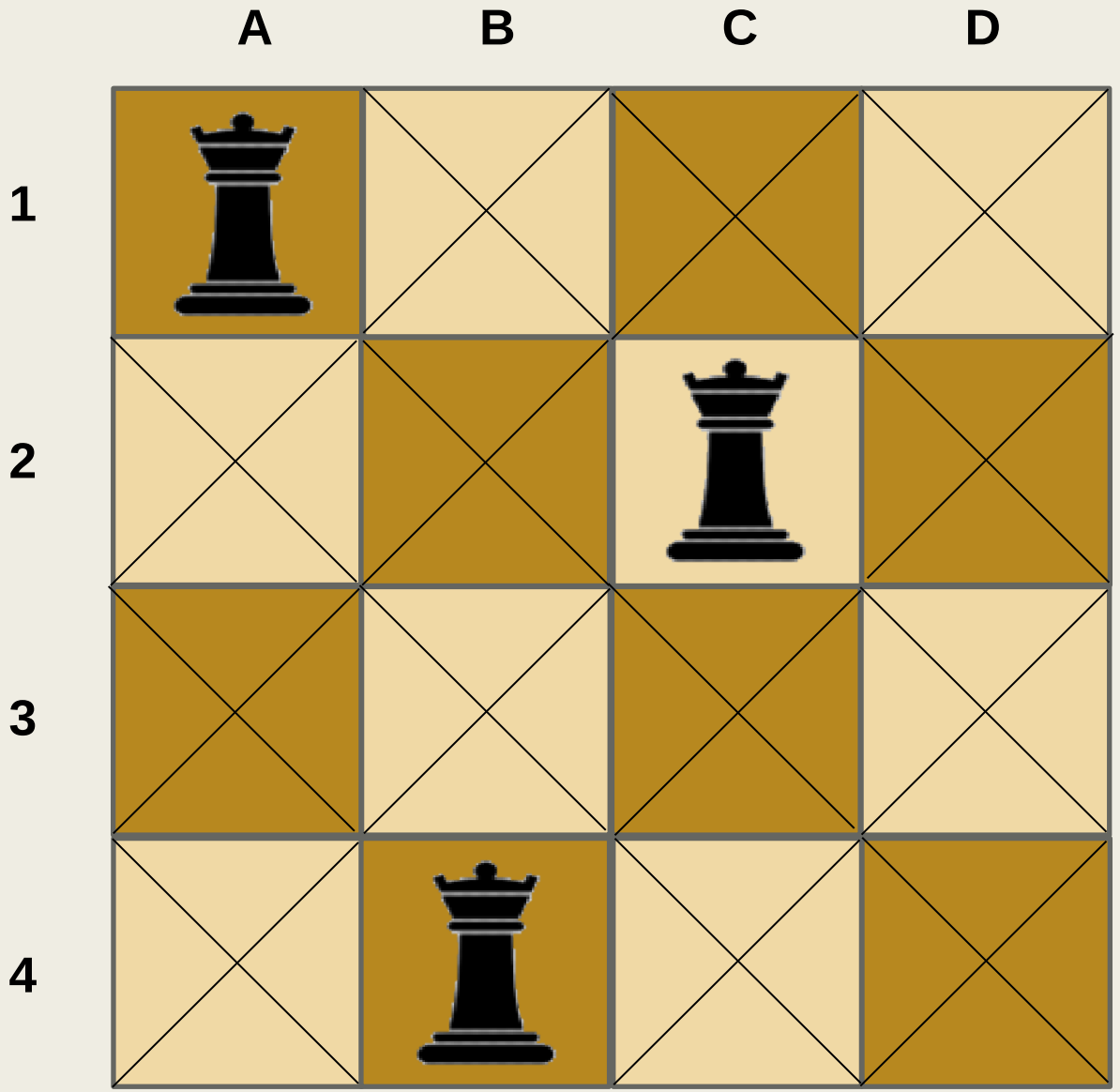
9663 N-Queen

N = 4



9663 N-Queen

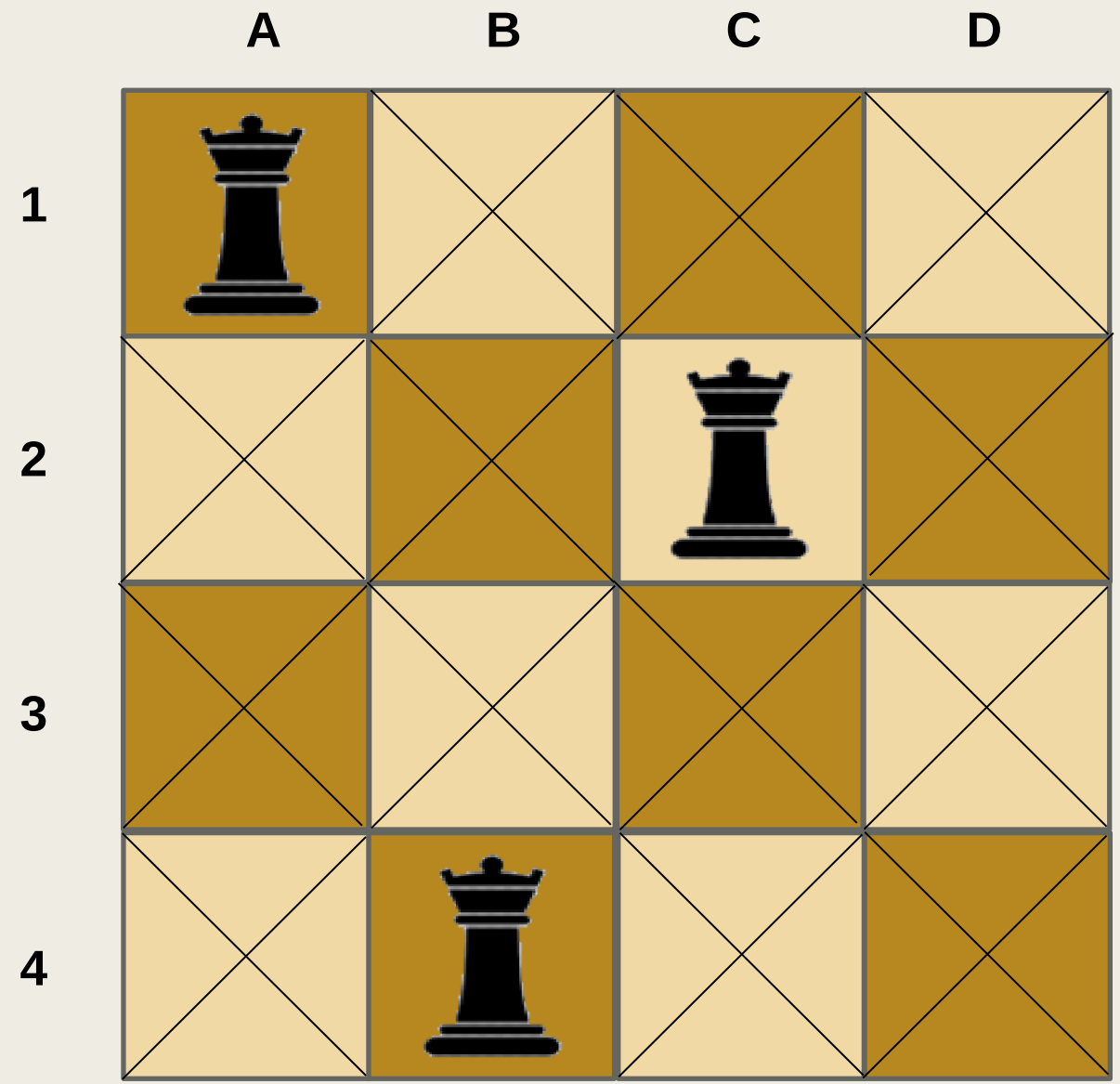
N = 4



9663 N-Queen

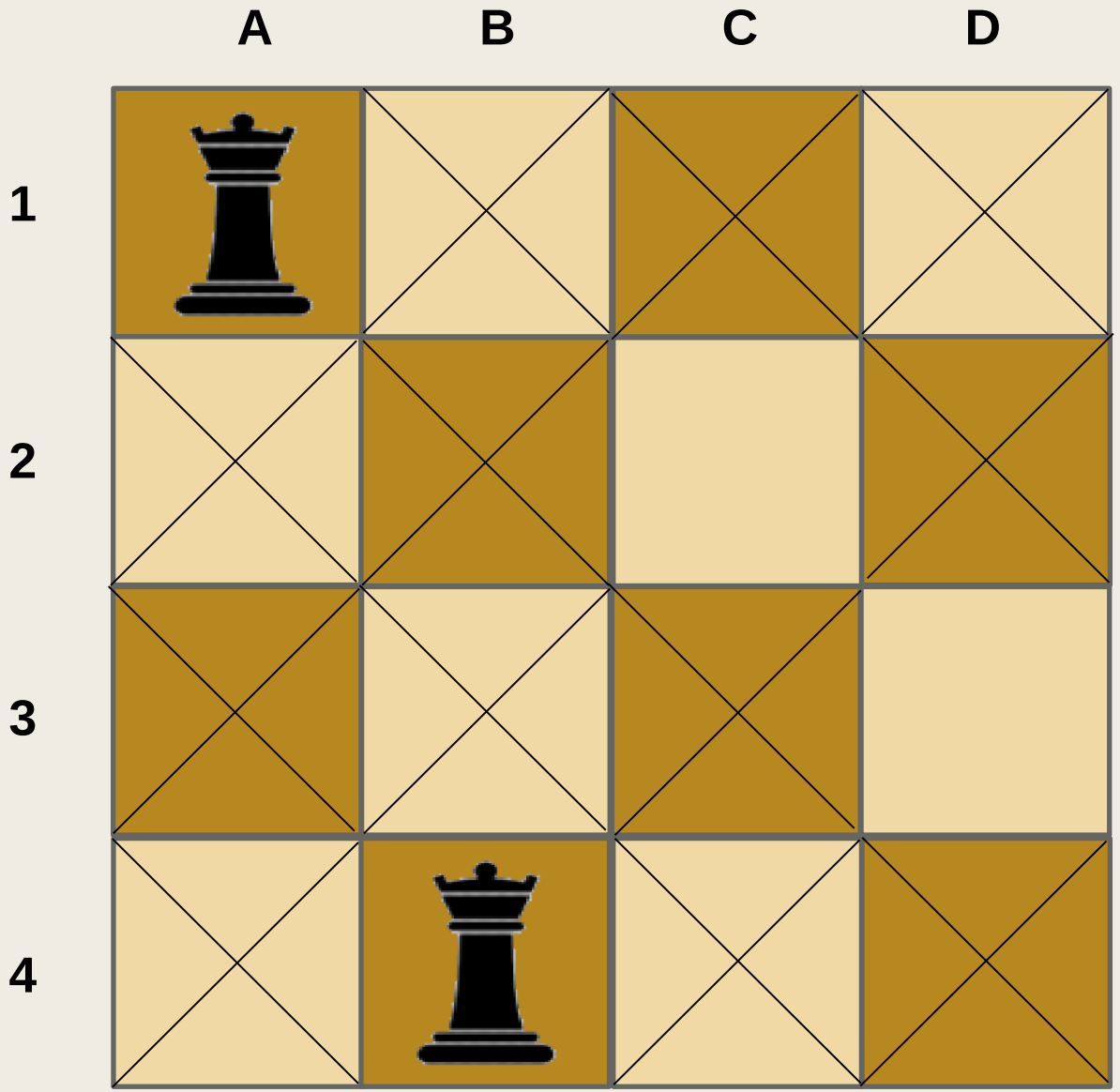
N = 4

4번째 퀸을 둘 자리가 없다!



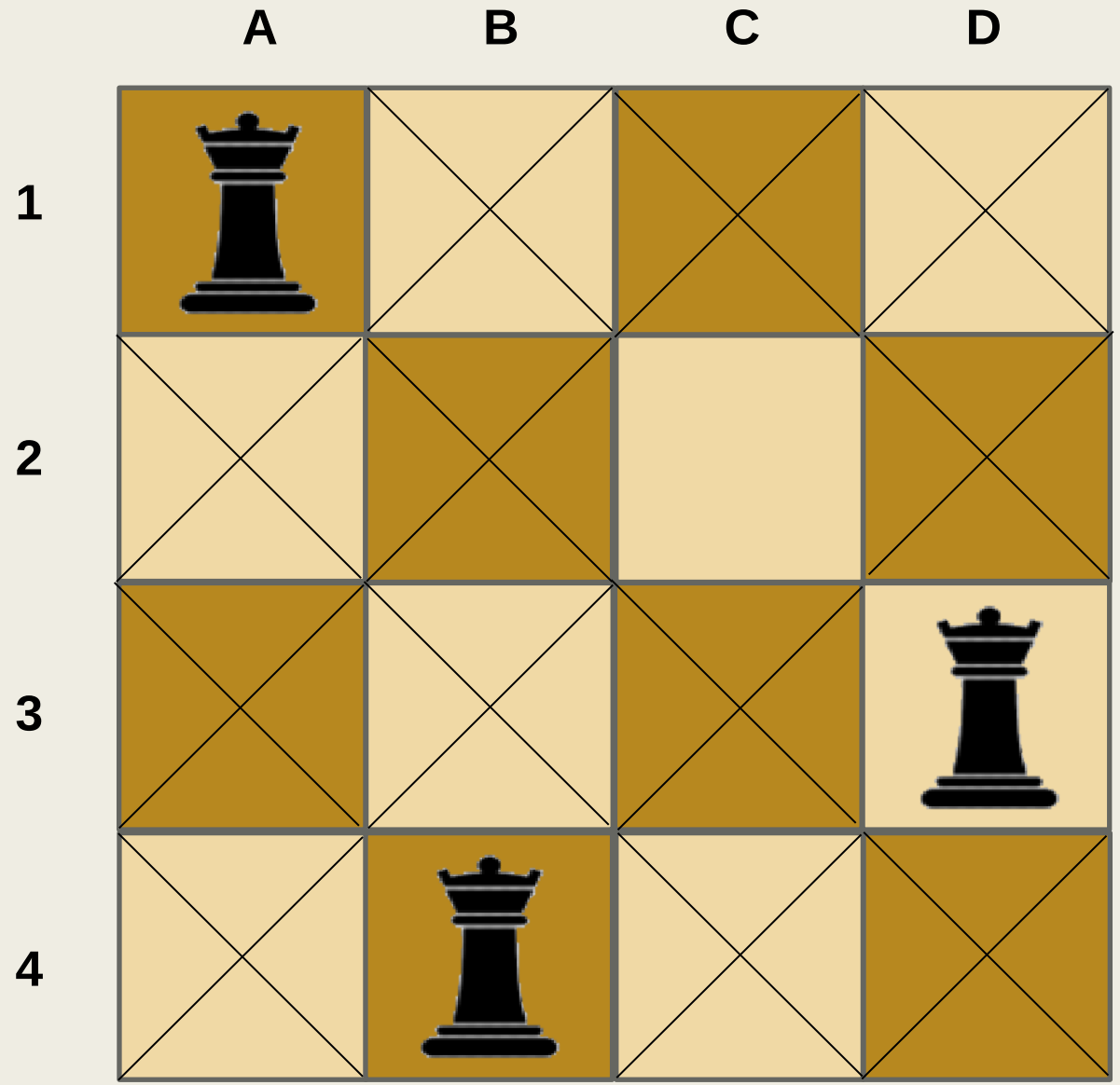
9663 N-Queen

N = 4



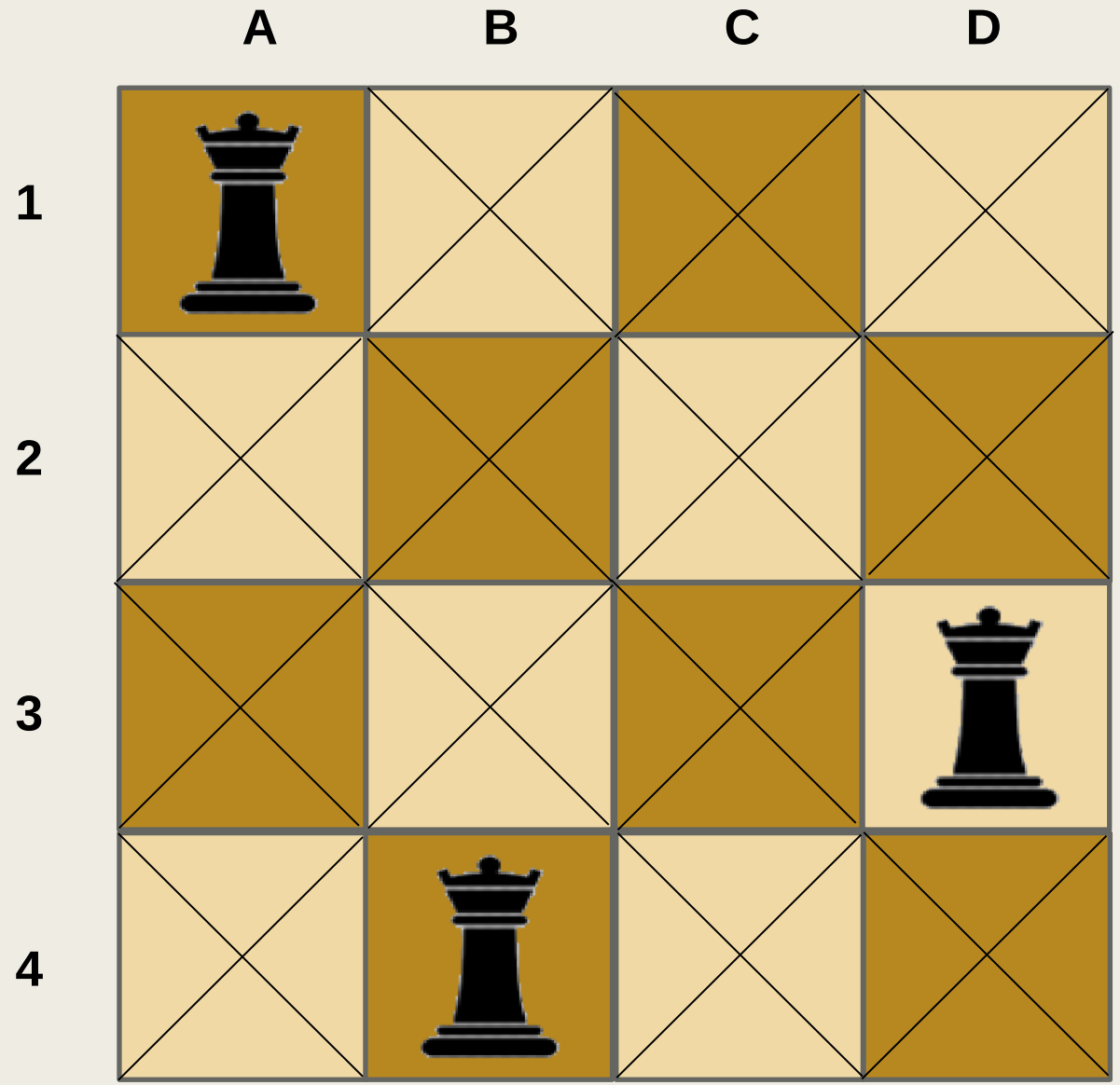
9663 N-Queen

N = 4



9663 N-Queen

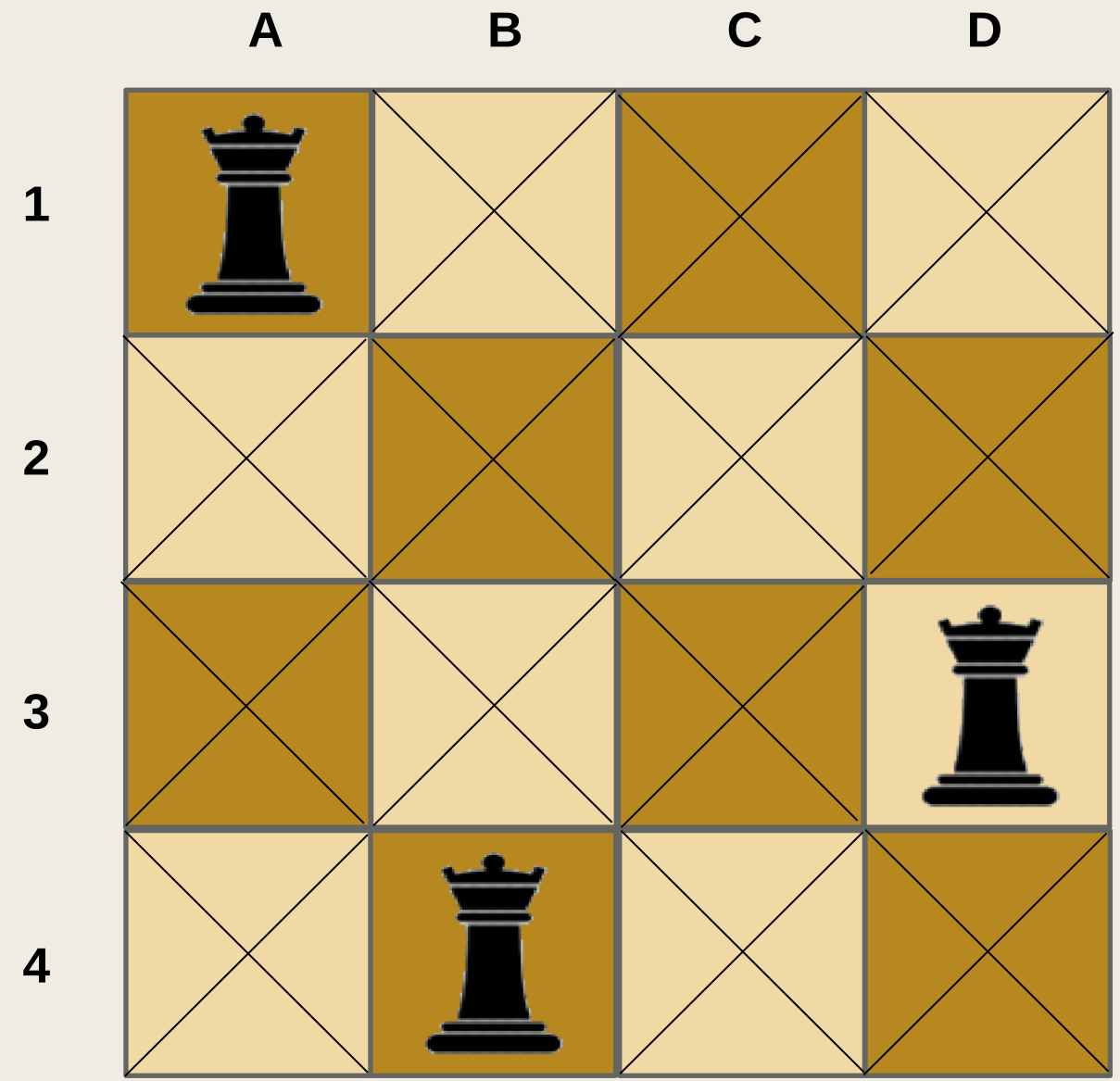
N = 4



9663 N-Queen

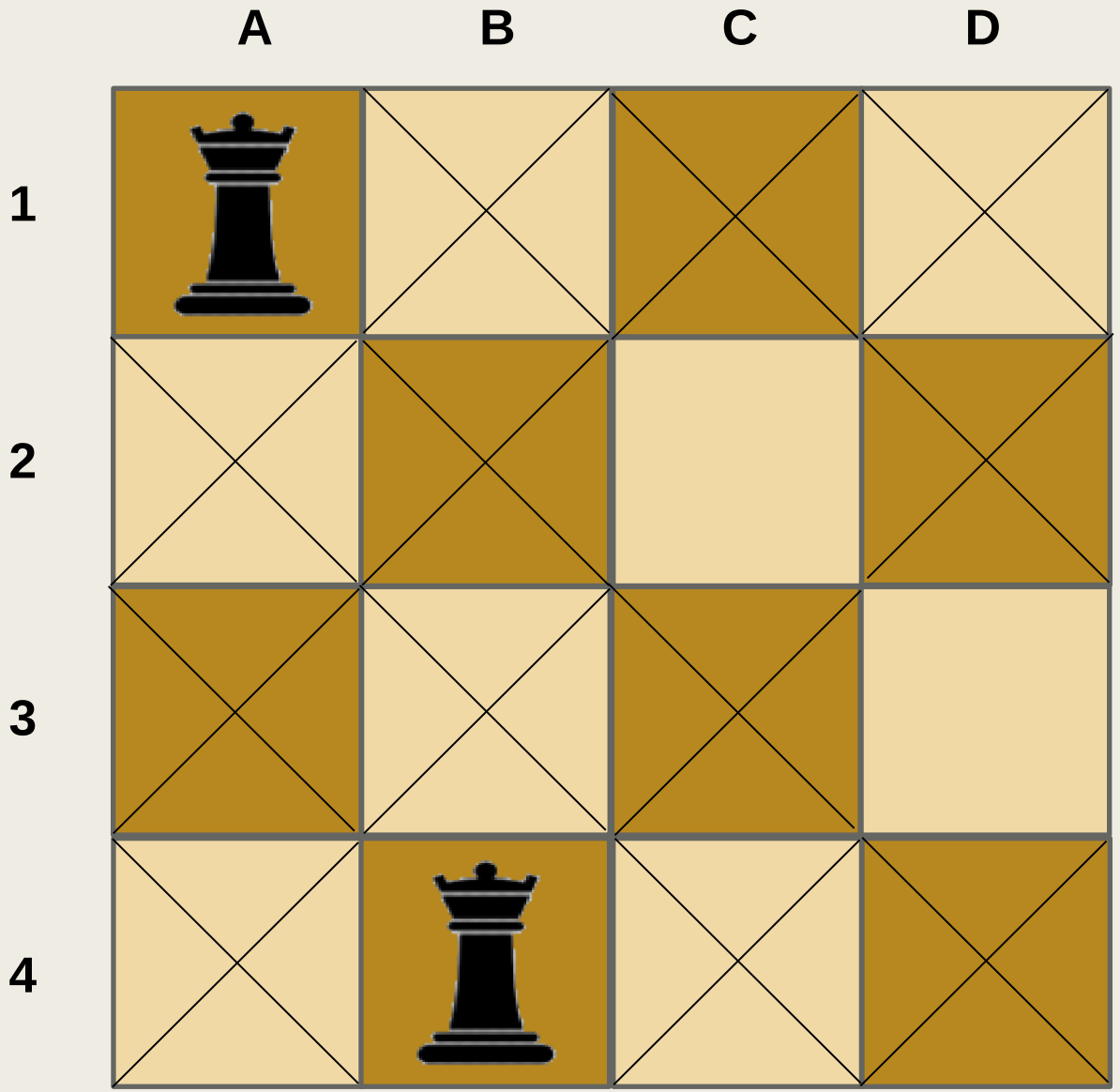
N = 4

4번째 퀸을 둘 자리가 없다!



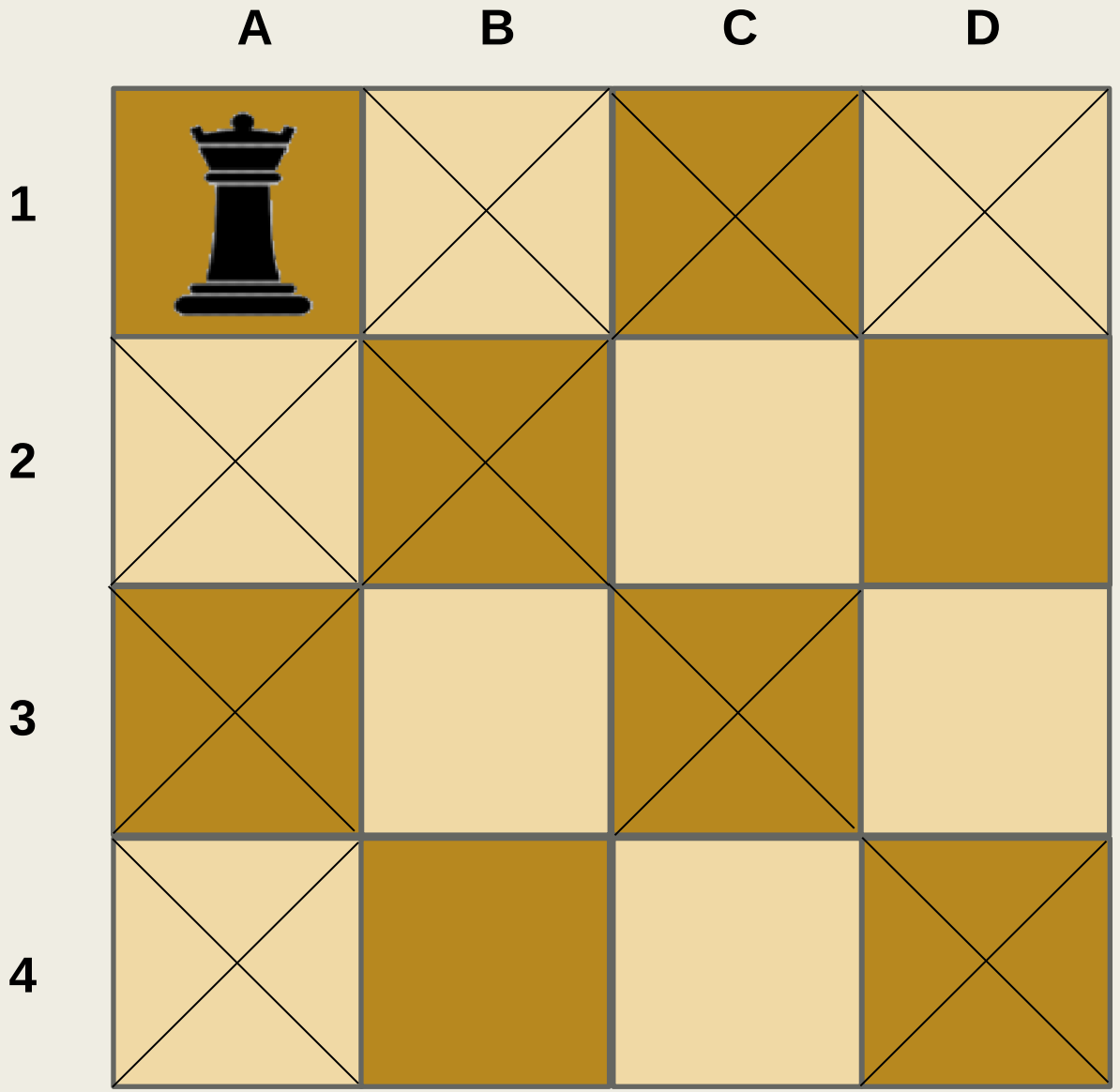
9663 N-Queen

N = 4



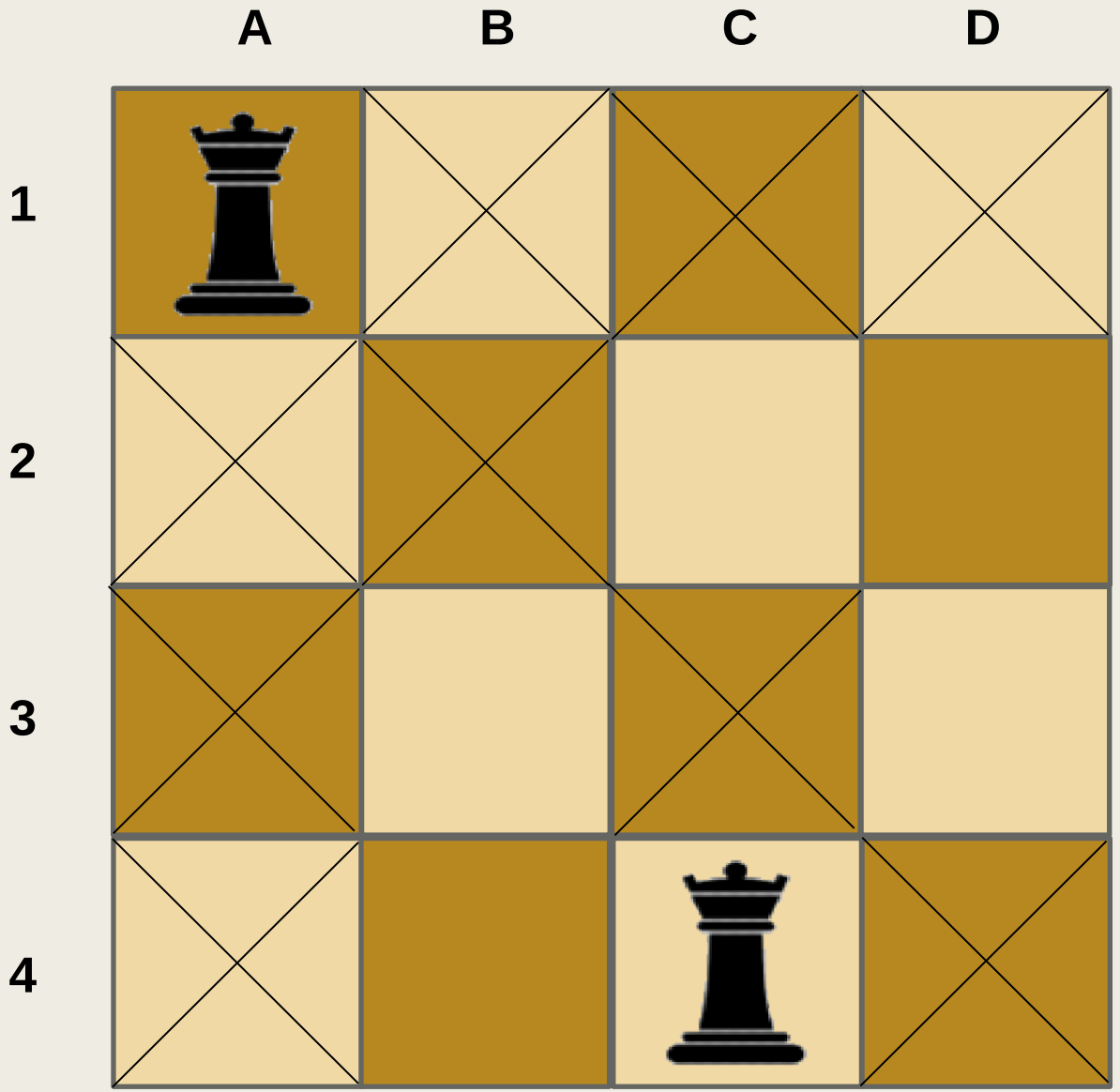
9663 N-Queen

N = 4



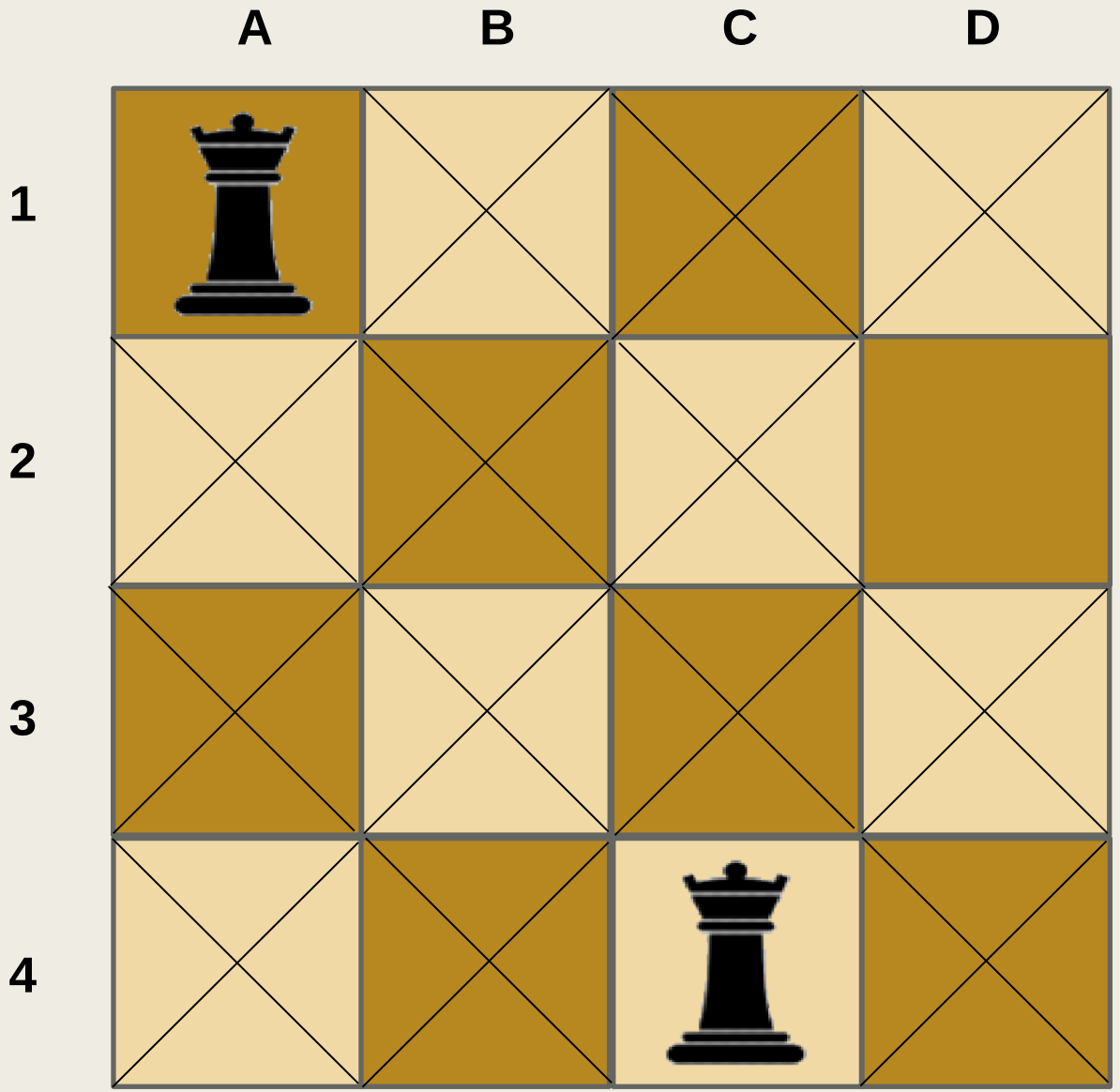
9663 N-Queen

N = 4



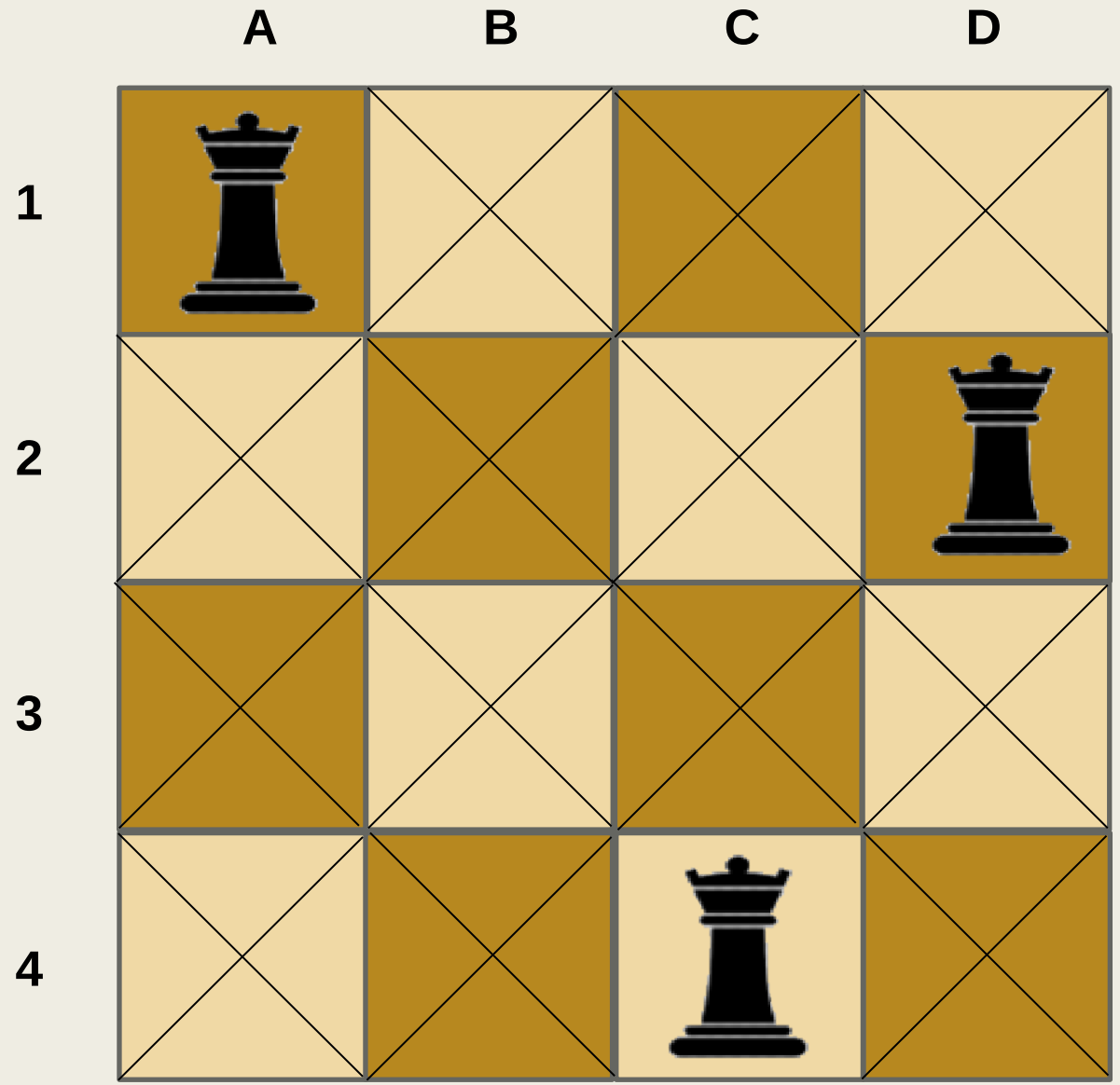
9663 N-Queen

N = 4



9663 N-Queen

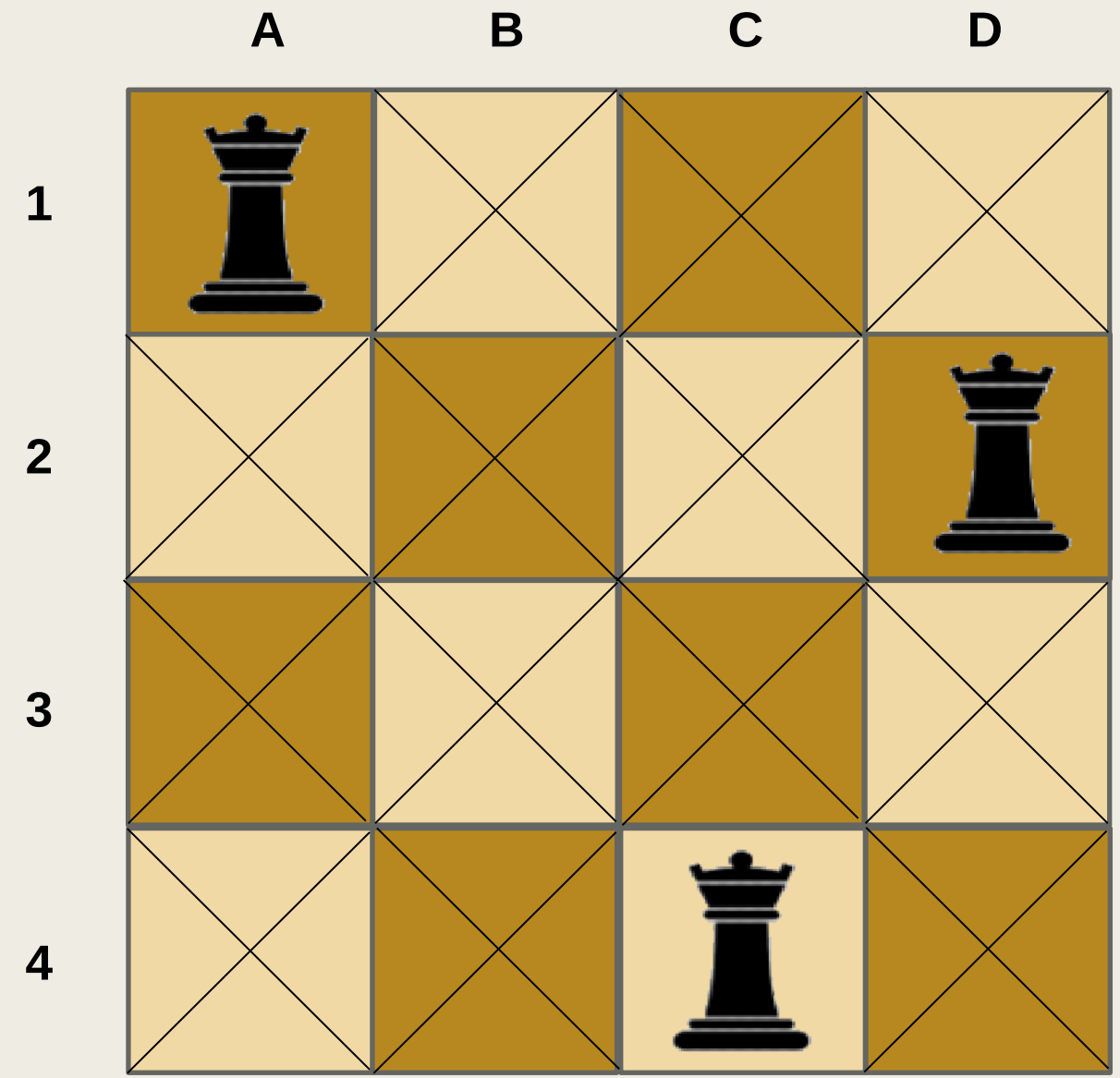
N = 4



9663 N-Queen

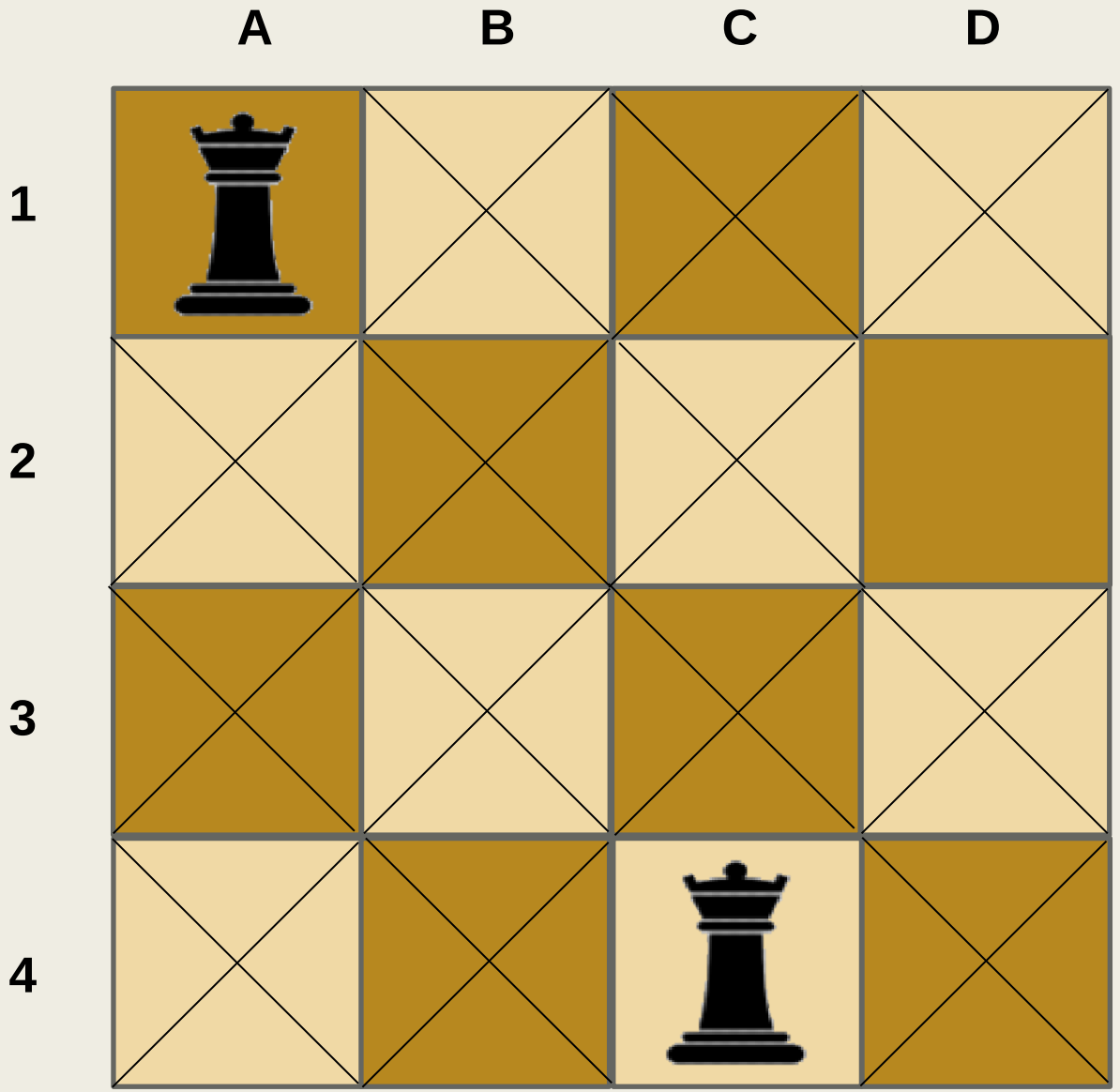
N = 4

4번째 퀸을 둘 자리가 없다!



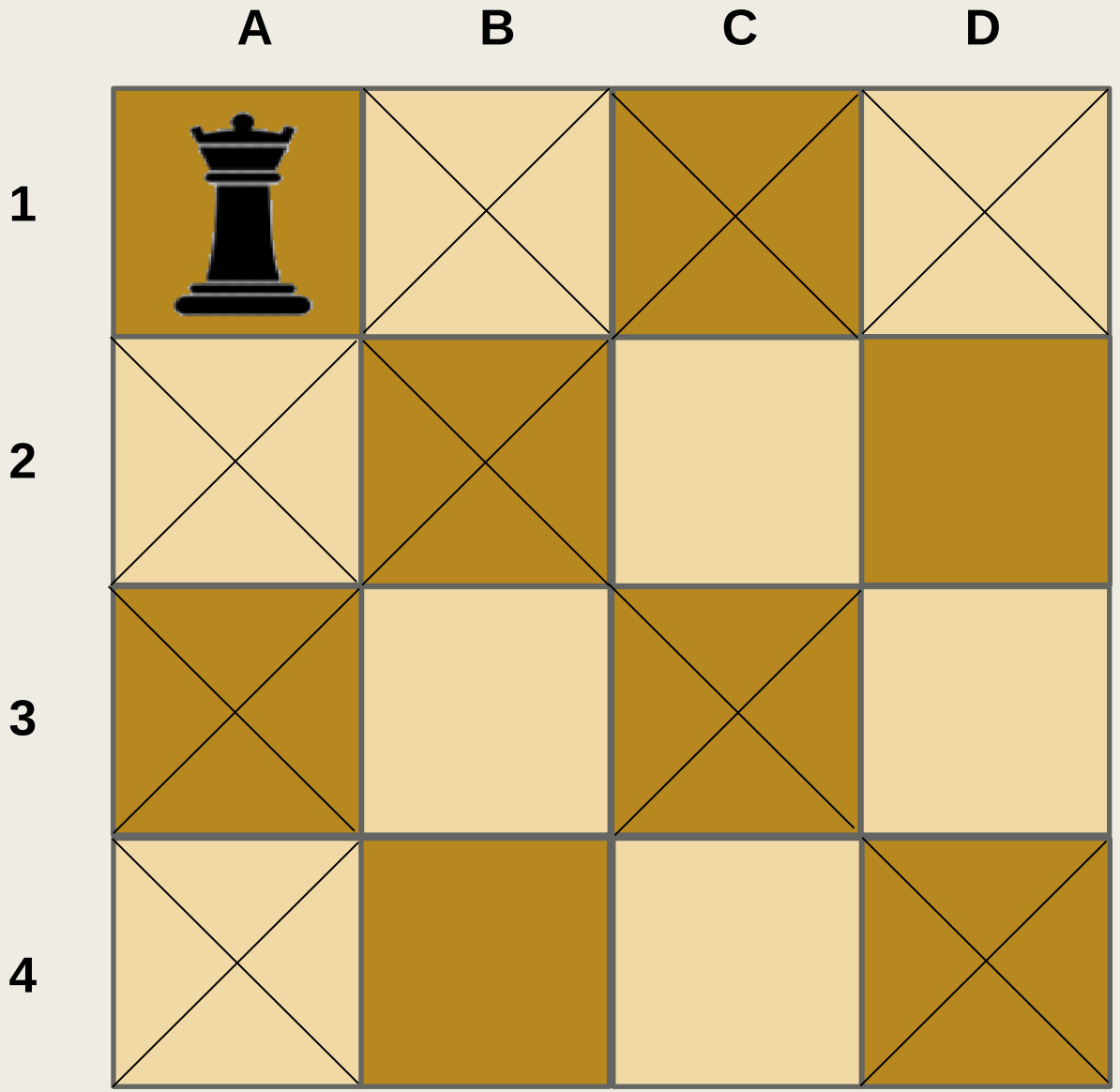
9663 N-Queen

N = 4



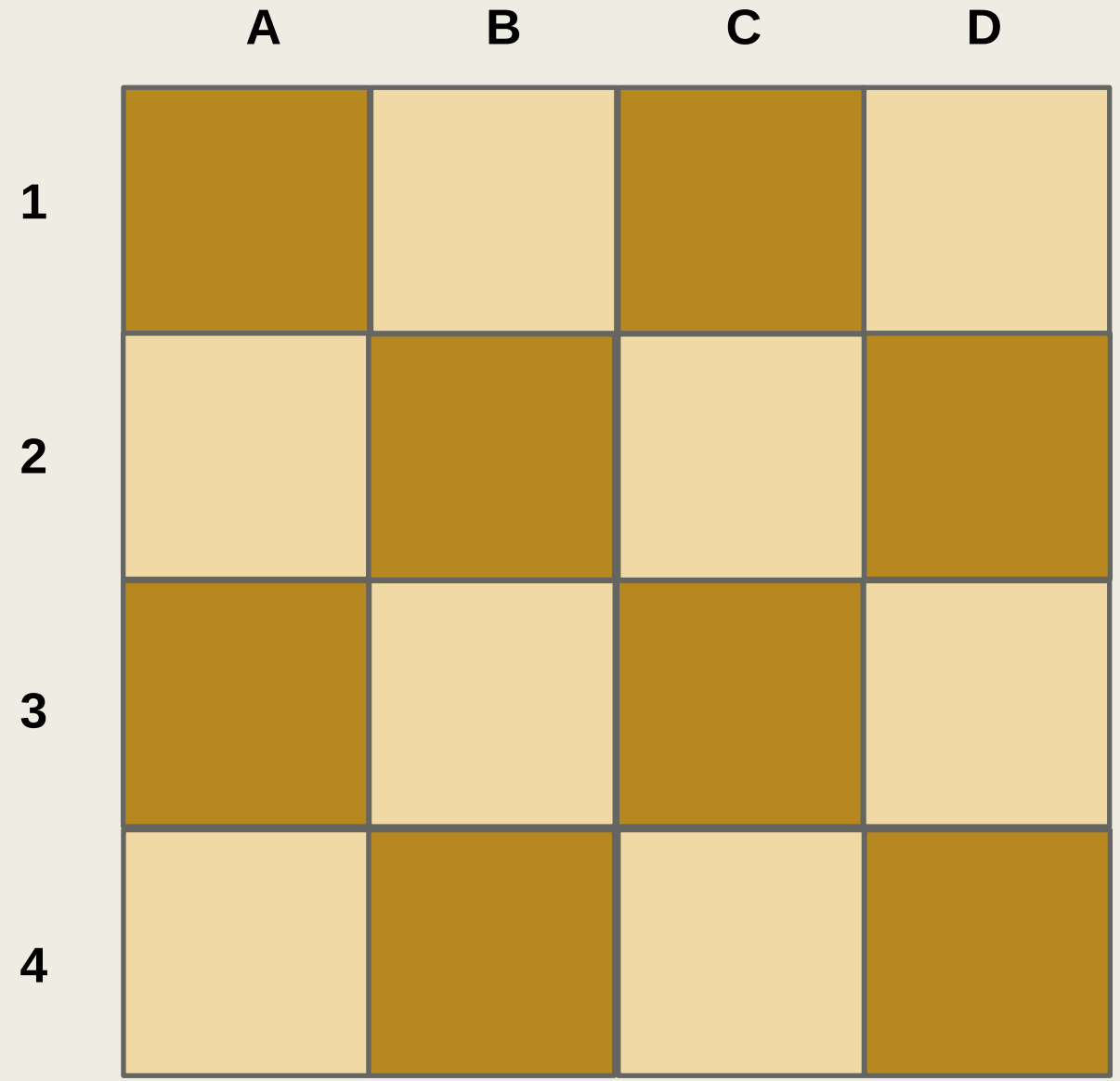
9663 N-Queen

N = 4



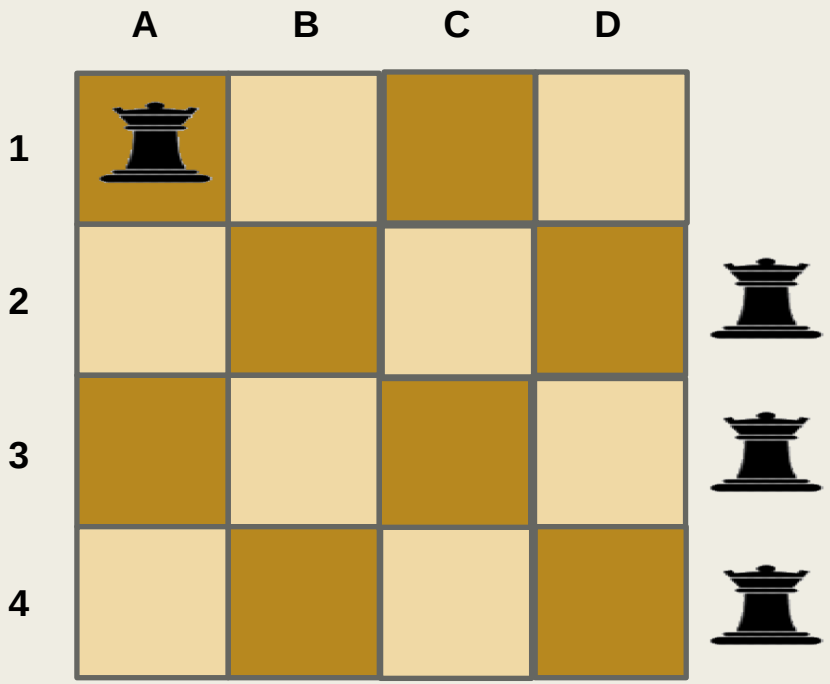
9663 N-Queen

N = 4



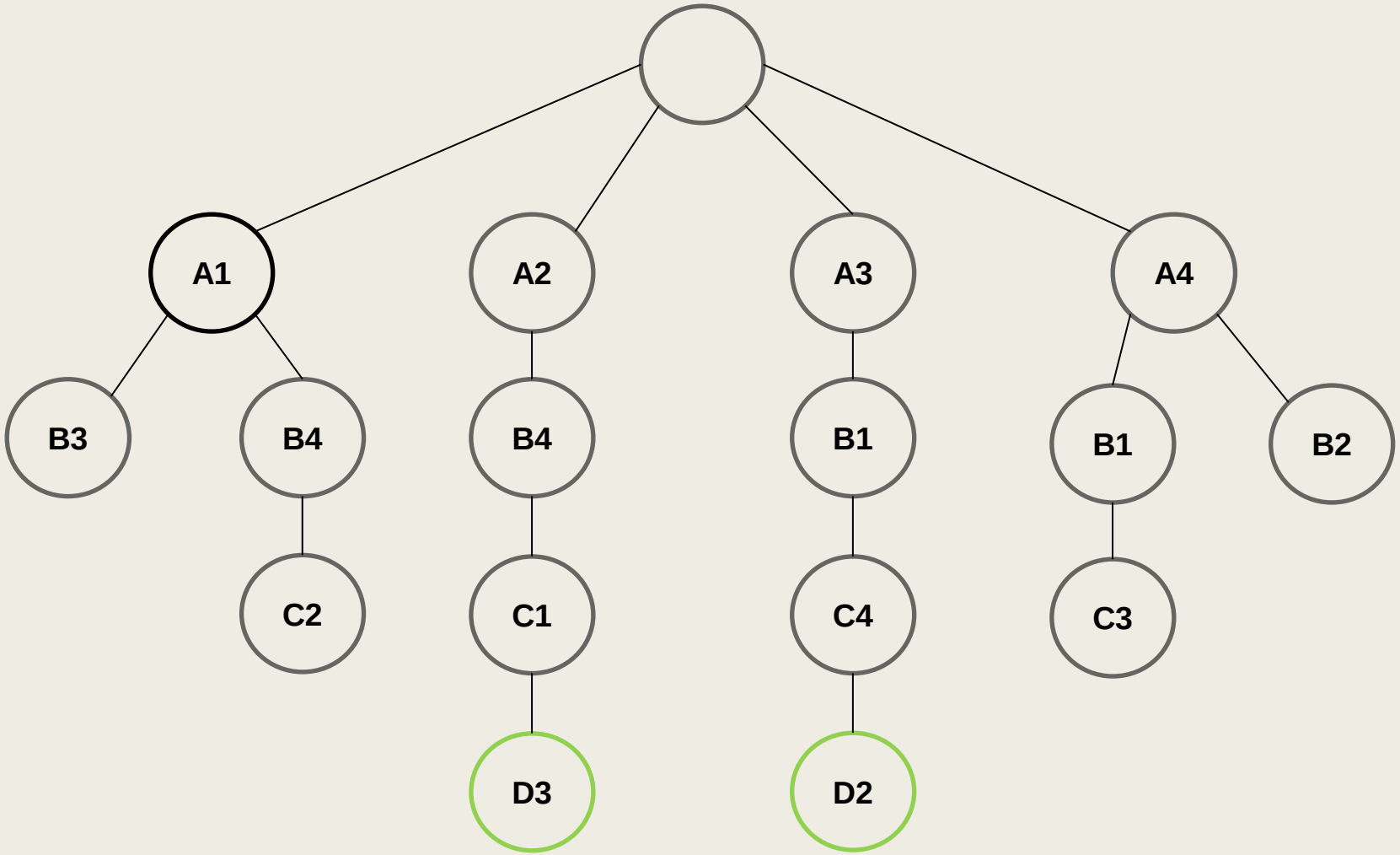
9663 N-Queen

N = 4



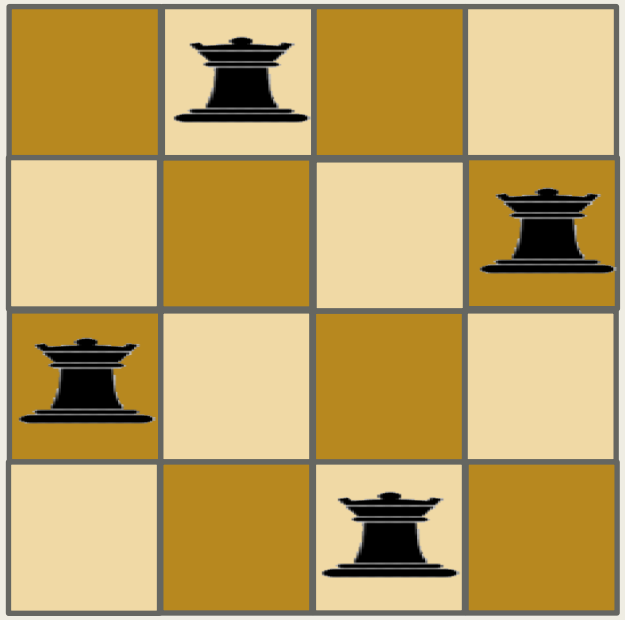
9663 N-Queen

N = 4

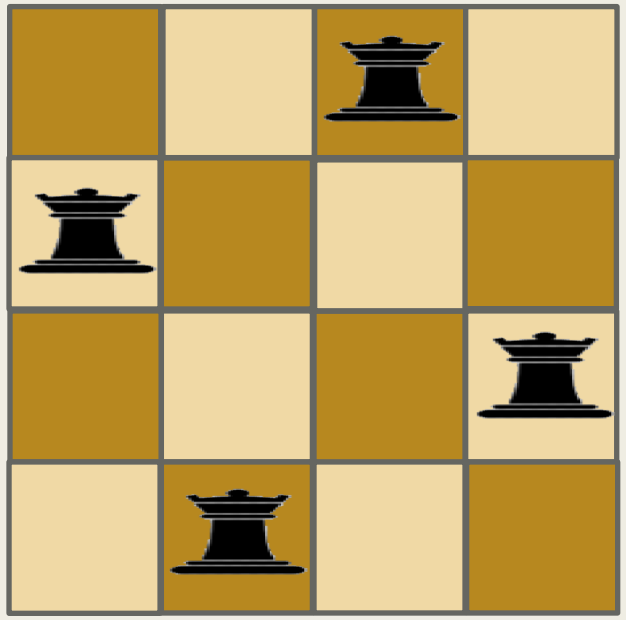


9663 N-Queen

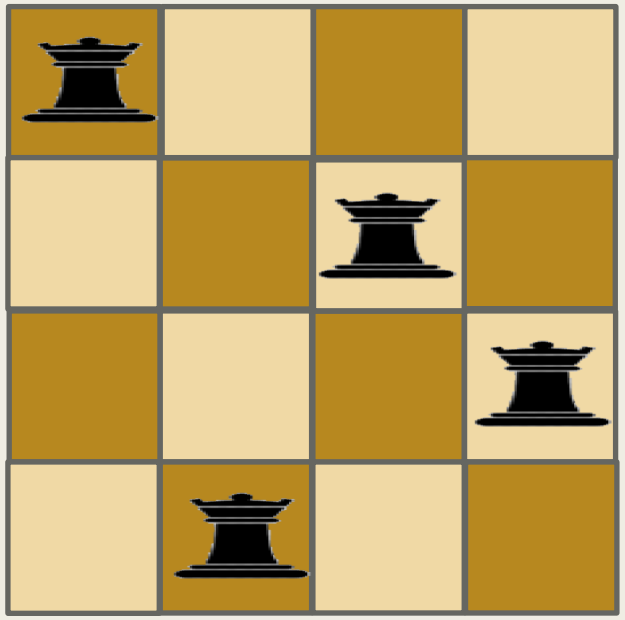
N = 4



가능한 배치 1



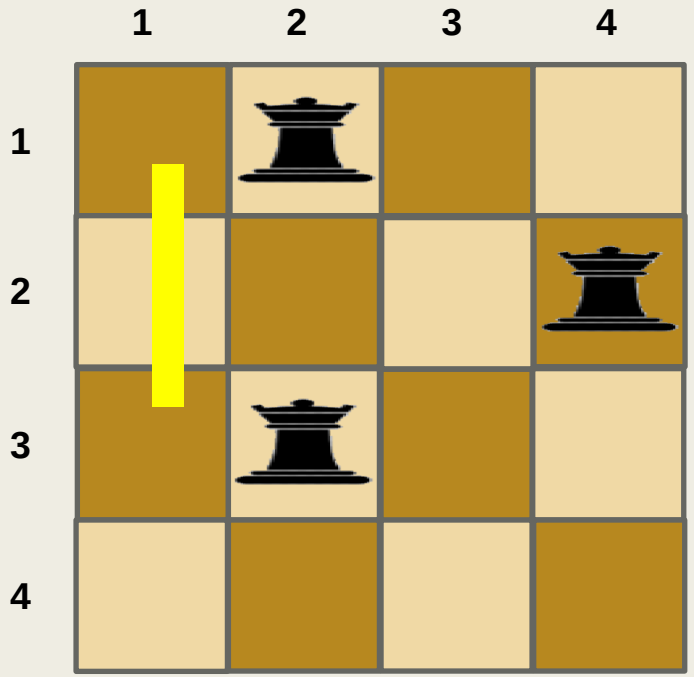
가능한 배치 2



불가능한 배치

9663 N-Queen

N = 4



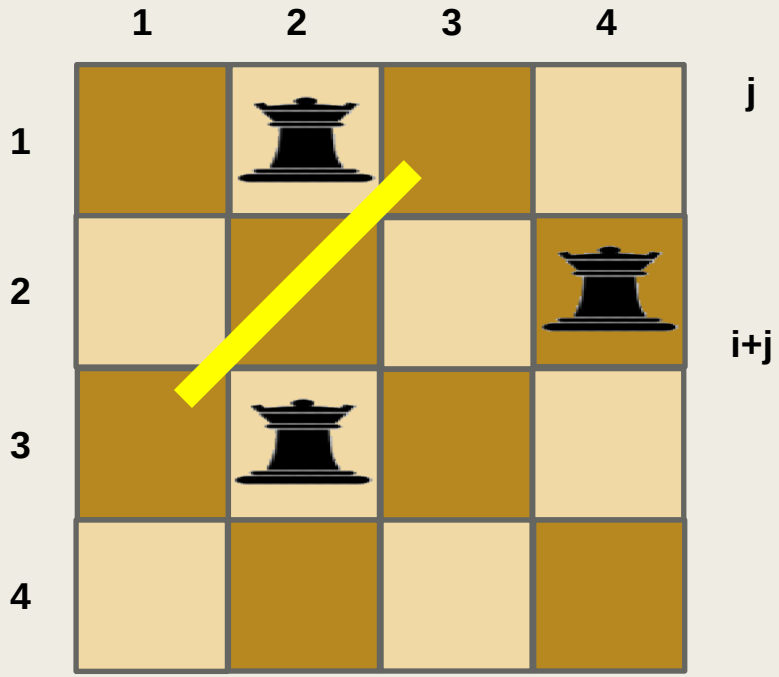
j

isused1

F	T	F	T

9663 N-Queen

N = 4

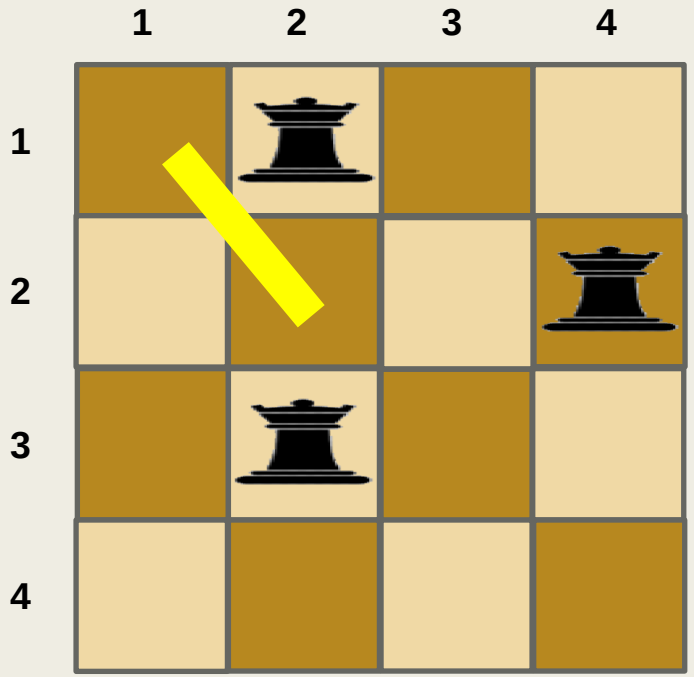


isused1	F	T	F	T

isused2	F	T	F	T	T	F	F

9663 N-Queen

N = 4



j

i+j

i-j+N-1

isused1	F	T	F	T

isused2	F	T	F	T	T	F	F

isused3	F	T	T	F	T	F	F

9663 N-Queen

```cpp
#include<iostream>
using namespace std;
#define MAX 40

int N;
bool isused1[MAX]; // 같은 열에 존재하는 퀸이 존재하는지 여부
bool isused2[MAX]; // 우상 대각선에 존재하는 퀸이 존재하는 여부
bool isused3[MAX]; // 좌상 대각선에 존재하는 퀸이 존재하는 여부
int cnt = 0; // 만족하는 배치 갯수 저장

void solve(int level) {
    if (level == N) {
        cnt++;
        return;
    }
    for (int i = 0; i < N; i++) {
        if (isused1[i] || isused2[i+level] ||
            isused3[level-i+N-1])continue;
        isused1[i] = true;
        isused2[i+level] = true;
        isused3[level-i+N-1] = true;
        solve(level + 1);
        isused1[i] = false;
        isused2[i + level] = false;
        isused3[level - i + N - 1] = false;
    }
}

int main() {
    cin >> N;
    solve(0);
    cout << cnt;
}
```

필수문제

- A - 분해합
- B - 체스판 다시 칠하기
- C - N과 M (1)
- D - N과 M (2)
- E - N-Queen

연습문제

- A - 부분수열의 합
- B - 스도쿠
- C - The Clocks
- D - Barbells
- E - N과 M (4)
- F - N과 M (5)
- G - N과 M (6)
- H - N과 M (7)
- I - N과 M (8)
- J - N과 M (9)
- K - N과 M (10)
- L - N과 M (11)
- M - N과 M (12)
- N - 복면산?!
- O - 마인크래프트
- P - 회의실 배정 2

피드백 및 질의응답

감사합니다!