1. **Finding Celebrities**

N명의 사람들 사이에서 Celebrity를 찾아내는 방법을 설계하는 문제이다.

Celebrity를 정의하면 다음과 같다.

1. 모두가 이 사람을 알고 있어야 한다.
2. 반대 이 사람은 다른 모든 사람들을 알지 못한다.

단순하게 한 사람 씩 비교해가면서 Celebrity를 찾으면 시간 복잡도가

비효율적이기 때문에 더 효율적인 방법을 찾아야 했다.

본인은 Celebrity 문제를 그래프 탐색 문제로 보고 풀어보았다.

각 사람들을 그래프의 노드로 보고, A가 B를 아는지 물어보는 것은

노드 간의 지향성 간선으로 보았다.

그 다음 한번에 한 명씩 제거해주면서 Celebrity 후보를 찾을 수 있는데,

예를 들어 두명의 사람 A, B가 비교하여, A가 B를 안다면 A는 Celebrity가 아니므로

A를 제외하고 그렇지 않다면 B가 Celebrity가 아니므로 B를 제거한다.

이 과정을 통해 한명의 Celebrity 후보가 남습니다.

마지막으로 이 후보가 진짜 Celebrity인지 확인하기 위해 모든 사람들이 그를 알고,

그는 모든 사람들을 모르는지 확인하면 된다.

두 명을 비교하여 한 명을 제외시키고, 제외되지 않은 사람과 다른 사람을 또 비교하는

방식을 취했기 때문에 시간 복잡도는 O(n)이며, 마지막에 진짜 Celebrity를 확인하는

과정도 한 번씩 쭉 모든 사람들을 확인하기 때문에 시간 복잡도가 O(n)이 된다.

따라서 본인이 설계한 알고리즘의 전체 시간 복잡도는 O(n)이다.

사람들의 수n과 각 사람과 다른 사람들 간의 모든 관계에 대해 저장해야 하기 때문에

공간 복잡도는 O(n2)이다.

1. **Minimum calculation**

a와 n이 주어졌을 때, b = an을 최소한의 곱셈 횟수로 계산하는 방법을

설계하는 문제이다.

일반적으로 an을 구하기 위해서는 n-1번의 곱셈이 필요하지만

더 효율적인 방법을 생각해보았다.

본인은 이 문제를 분할 정복 방법을 통해 풀어보았다.

예를 들어 n이 짝수일 때는 로 표현할 수 있으며,

n이 홀수일 때는 로 표현한다.

이 표현과 재귀를 이용하면 빠르게 계산할 수 있다.

예시: n = 16일 때,

시작 🡪 a16

재귀 🡪 a8

재귀 🡪 a4

재귀 🡪 a2

재귀 🡪 a

총 네 번의 재귀를 통해서 a16을 구할 수 있다.

이 방법을 이용하면 시간 복잡도가 O(log n)이며,

재귀를 통해 스택 메모리를 사용하므로 공간 복잡도는 O(log n)이다.