

Hanoi Tower

Hanoi Tower 문제는 아래 그림과 같이 3 개의 기둥과 여러 개의 크기(지름)가 다른 원반으로 구성되어 있는데, 제일 처음에는 가장 왼쪽 기둥에 크기가 큰 원반부터 가장 작은 원반까지 아래부터 위로 원뿔모양으로 쌓여있다.

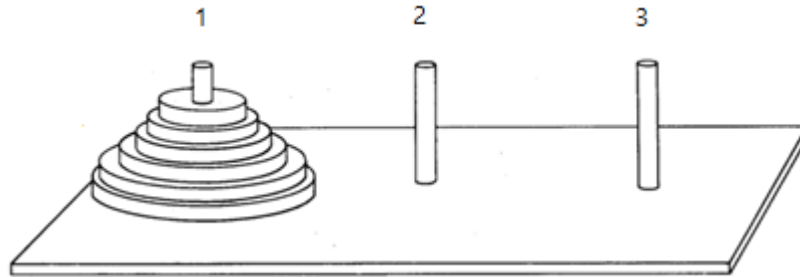


그림 1

(figure by Neapolitan & Naimipour)

Hanoi Tower 문제는 가장 왼쪽 기둥에 쌓여 있는 모든 원반을 가장 오른쪽 기둥으로 다음 조건을 만족시키면서 모두 옮기는 것이다.

- (1) 한 번에 한 개의 원반만을 옮길 수 있다.
- (2) 한 개의 원반을 옮길 때는 어떤 기둥에 쌓여있는 가장 위쪽의 원반을 꺼내어 옮겨놓고자 하는 다른 기둥으로 옮긴다. 이 때 옮겨놓고자 하는 기둥에 쌓여 있는 원반들의 가장 위에만 놓을 수 있다. 혹은 그 기둥에 원반이 없을 경우에는 그 기둥의 가장 아래에 놓여지게 된다.
- (3) 원반은 자신보다 작은 원반위에 놓여질 수 없다.

위 그림에서 가장 왼쪽 기둥을 1 번, 중간 기둥을 2 번, 오른쪽 기둥을 3 번이라 하고, 1 번 기둥에 쌓여 있는 모든 원반을 3 번 기둥으로 옮기고자 한다. 예를 들어, 3 개의 원반인 경우에는 아래 그림과 같이 7 번 원반을 옮겨서 문제를 해결할 수 있다.

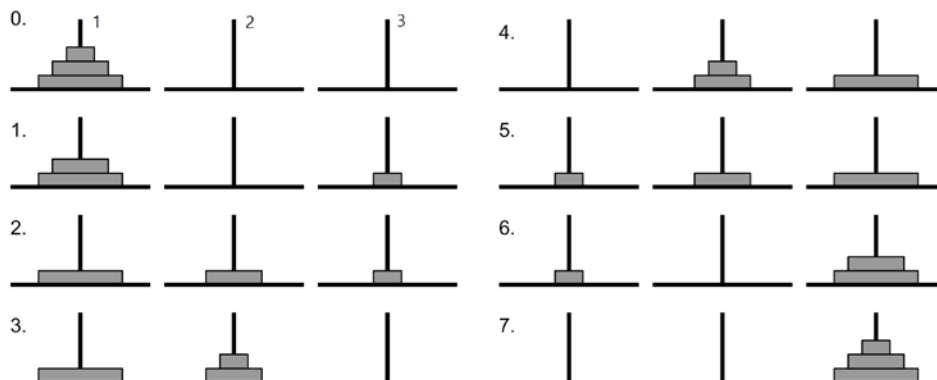


그림 2

Hanoi Tower 문제는 아래 그림과 같이 recursive 하게 문제를 해결할 수 있다. 즉 n 개의 원반이 있을 때,

- (1) 1 번 기둥에 쌓여 있는 원반들 중에 가장 아래쪽에 놓여져 있는 가장 큰 원반을 제외한 나머지 $n-1$ 개의 원반들을 2 번 기둥으로 옮긴다 (recursion).
- (2) 1 번 기둥에 있는 가장 큰 원반을 3 번 기둥으로 옮긴다.
- (3) 2 번 기둥으로 옮겨 놓았던 $n-1$ 개이 원반들을 모두 3 번 기둥으로 옮긴다 (recursion).

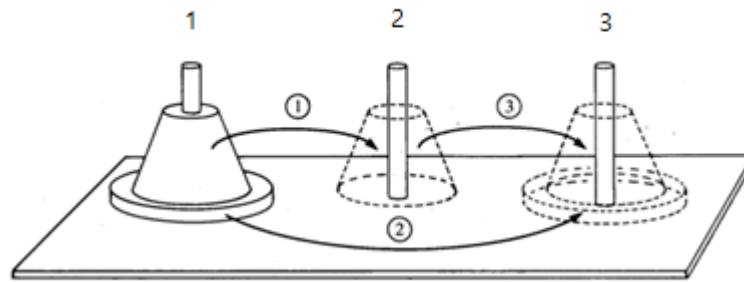


그림 3

(figure by Neapolitan & Naimipour)

위의 작업을 pseudo-code 로 작성하면 아래와 같다.

```
// n: number of disks, a: source peg, c: target peg, b: intermediate peg
algorithm hanoi_tower(n, a, b, c)
    if  $n > 0$  then
        hanoi_tower(n-1, a, c, b);
        print a, " -> ", c
        hanoi_tower(n-1, b, a, c);

procedure solve_hanoi_tower()
    num_disks = 3;
    Hanoi_tower(num_disks, 1, 2, 3);
```

위의 예에서와 같이 3 개의 원반으로 구성된 Hanoi Tower 문제의 출력은 아래와 같다 (그림 2 참조).

```
1 -> 3
1 -> 2
3 -> 2
1 -> 3
2 -> 1
2 -> 3
1 -> 3
```

Hanoi Tower 문제가 n 개의 원반으로 구성되어 있으며, 이 문제를 위의 pseudo-code 로 해결하고자 할 때, k ($1 \leq k < 2^n - 1$) 번째로 옮겨지는 원반의 source 기둥 (그 원반을 꺼낼 기둥) 번호와 target 기둥 (그 원반을 옮겨 놓을 기둥) 번호를 출력하는 프로그램을 작성하시오.

예를 들어, 3 개의 원반이 주어졌을 때, 5 번째로 옮길 원반의 source 기둥 번호는 2 번이며, target 기둥 번호는 1 번이다.

입력

입력은 표준입력(standard input)을 사용한다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 t 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 정수들이 주어진다. 각 테스트 케이스에 해당되는 각 줄에 두개의 정수 n, k 가 주어지는데, 첫 번째 정수 n ($1 \leq n \leq 64$) 은 Hanoi Tower 문제에서 주어지는 원반의 개수이며, 그 다음 정수 k ($1 \leq k < 2^n - 1$) 는 k 번째 옮길 원반을 나타낸다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 k 번째로 옮겨질 원반의 source 기둥 번호와 target 기둥 번호를 출력한다. 두 정수 사이에는 한 개의 공백을 둔다.

입력과 출력의 예

입력
3
3 5
1 1
64 18446744073709551615

출력
2 1
1 3
2 3

(주의)

1. 정수 type 을 64-bit “unsigned long long int” 을 사용하기 바람.