

[BOJ] 1914번: 하노이 탑

문제 접근

- n 이 20보다 클 때 과정 생략
- 하노이 탑의 원판 이동 횟수 = $2^n - 1$
- n 이 커질수록 이동 횟수 매우 커지므로 BigInteger를 사용
- 하노이 탑은 재귀 호출


BigInteger

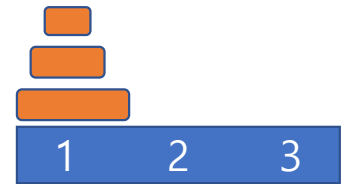
- java에서 아주 큰 정수를 다룰 수 있는 클래스.
- 문제에서 n 의 범위가 `부터 100까지이었는데, 만약 n 이 100일 경우에
- $2^n - 1$ 은 매우 큰 수가 되기 때문에 int나 long이 아닌 BigInteger를 사용한다.


```
29
30 public static int hanoi(int from, int to, int n) {
31     if(n == 1) { // 원판이 1개일 경우에는 가장 작은 원판을 from기둥에서 to 기둥으로 바로 옮긴다.
32         sb.append(from + " " + to).append("\n");
33         return 0;
34     }
35     // n이 1이 아닌 경우에는 임시기둥으로 'n-1'개의 원판을 옮긴다.
36     // 임시기둥이 6에서 빼는 이유는 하노이 탑의 기둥 번호가 1,2,3으로 주어지기 때문이다.
37     hanoi(from, to: 6 - from - to, n: n - 1);
38     sb.append(from + " " + to).append("\n"); // 가장 큰 원판을 from -> to 기둥으로 옮긴다.
39     hanoi(from: 6 - from - to, to, n: n - 1); // 임시기둥에 옮겨뒀던 'n-1'개의 원판을 to로 옮긴다.
40     return 0;
41 }
42 }
```

hanoi(from, to, n)


30: hanoi(1, 3, 3)

↳ 37: hanoi(1, 2, 2) 

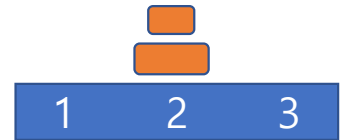


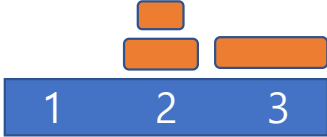
↳ 37: hanoi(1, 3, 1) 

➡ 31: 기둥 1 -> 기둥 3으로 가장 작은 원판(1) 이동(기록)

38: 기둥 1 -> 기둥 2로 가장 큰 원판(2) 이동(기록) 

39: hanoi(3, 2, 1)
➡ 31: 기둥 3 -> 기둥 2으로 가장 작은 원판(1) 이동(기록)



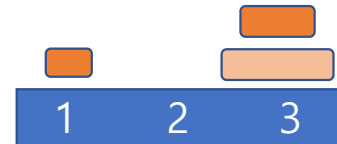
38: 기둥 1 -> 기둥 3으로 가장 큰 원판(3) 이동(기록) 

39: hanoi(2, 3, 2)

↳ 37: hanoi(2, 1, 1)
➡ 31: 기둥 2 -> 기둥 1으로 가장 작은 원판(1) 이동(기록)



38: 기둥 2 -> 기둥 3으로 가장 큰 원판(2) 이동(기록)



39: hanoi(1, 3, 1)

➡ 31: 기둥 1 -> 기둥 3으로 가장 작은 원판(1) 이동(기록)

