1. 문제 관찰

\* 기둥 1, 2, 3이 있고, 원판이 N개가 있습니다.

\* 기둥 1에서 3까지 모든 원판을 옮겨야 합니다.

\* 기둥 1에서 3까지 옮기기 위해서는 n-1개의 원판을 우선 기둥 2로 옮겨야 합니다.

\* 이후, 기둥 1에 남아있는 n번째 원판을 3으로 옮깁니다.

\* 이후, 기둥 2에 있던 n-1의 원판을 기둥 3으로 옮깁니다.

\* 문제는 이 과정을 모두 출력해야 합니다.

\* 이 문제는 다음과 같은 재귀적사고로 생각할 수 있습니다.

원판 1(개)을 기둥 1에서 3까지 옮길 수 있다.

원판 n(개)를 기둥 1에서 3까지 옮길 수 있다면, 원판 n+1(개)를 기둥 1에서 3까지 옮길 수 있다.

\* 원판 n개를 기둥 3(end\_point)까지 옮기려면 n-1개를 기둥 2(end\_point)까지 옮겨야 합니다. 고로 재귀의 end\_point가 그때 그때 달라질 수 있습니다.

\* 기둥 1(start\_point)에 있던 n번째 원판을 3으로 옮긴 후에는 기둥 2(start\_point)에 남아있던 n-1개까지의 원판을 기둥 3으로 옮겨야 합니다. 고로 재귀의 start\_point가 그때 그때 달라질 수 있습니다.

\* 재귀는 n == 1일 때, base condition이 됩니다.

\* 옮기는 순서를 출력해야 하기 때문에, 재귀가 base condition에 부딪혔을 때부터 출력이 돌아가게끔 코딩해야 합니다.

1. 알고리즘

Def Hanoi(start\_point, end\_point, n):

If n == 1:

Print(start\_point, end\_point)

Else :

Hanoi(start\_point, 6-start\_point-end\_point, n-1)

Print(start\_point, end\_point)

Hanoi(6-start\_point-end\_point, end\_point, n-1)