Assignment2 Document

#자기정보

|  |  |  |  |
| --- | --- | --- | --- |
| 전공 | 소프트웨어전공 | 학번 | 2015004011 |
| 이름 | 김도현 | 개발환경 | Xcode |

#과제 설명

하노이탑에서 기둥이 4개 일때 원반이 n개 있다고 했을때

1. 한번에 하나의 원판만 옮길 수 있다.
2. 큰 원판이 작은 원판 위에 있어서는 안된다.

이 두가지의 규칙에 의거하여 원반 하나를 mid1에 옮기게 되면 그 기둥에는 나머지 원판을 옮길 수 없다. 즉, 기둥 3개의 하노이탑의 문제가 된다.

이를 이용하면 n개중 i개를 mid에 옮기고나면 n-i개의 원판을 to로 옮기는 문제는 기둥 3개의 문제가 되는것이므로 n-i개를 to로 옮기고 middle에 옮겨둔 i개를 to로 옮기면 된다.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 기둥 3개 | 기둥 4개 | 기둥 5개 |
| 원반 3개 | 7 | 5 | 5 |
| 원반 5개 | 31 | 13 | 11 |
| 원반 10개 | 1023 | 49 | 31 |
| 원반 15개 | 32767 | 129 | 71 |
| 원반 20개 | 1048575 | 289 | 111 |

따라서, i개를 조절하여 이동횟수의 값을 모두 구한 후 최솟값을 구하는 방식을 사용했다.

최소 이동횟수만을 구하기 위해서 각 함수에 원판의 크기만 전달하도록 기존 코드를 수정하였고, 빠른 계산을 위하여 배열에 미리 값을 저장해두는 방식을 이용하였다.

그리고 메모리를 절약하기 위해 원판의 크기만큼 동적할당 하였고 똑같은 계산은 \*2를 해주었다.

int를 unsigned int로 수정하거나 long 같은 더 큰 크기의 타입을 사용한다면 더 많은 원판의 이동 횟수를 구할 수 있을 것으로 보인다.

#실행화면



