CHAC 2023 풀이

Official Solutions

by

중앙대학교 알고리즘 학회 ChAOS

	문제명	난이도
A	찬반투표	Easy
B	버섯 농장	Normal
C	연산자 파티	Normal
D	수열 재배열	Normal
E	시프트 연산	Hard
F	견제 미로찾기	Hard
G	월드컵 조별리그	Hard
H	어려운 하노이 탑	Challenging

A. 찬반투표

#Implementation
출제진 의도- Easy

✅ 출제자: njw1204
✅ 제출 16번, 정답 10팀 (정답률 62.5%)
✅ 처음 푼 팀: KANKAN, 2분

A. 찬반투표

- ✓ 투표 내역을 하나씩 입력 받으며 찬성, 반대, 기권의 표 수를 각각 변수에 저장합니다.
- ✓ 만약 (기권표의 수)×2가 N 이상이라면 투표가 무효처리 되므로 “INVALID”를 출력합니다.
- ✓ 그렇지 않고 만약 찬성의 개수가 반대의 개수보다 많다면 “APPROVED”를 출력합니다.
- ✓ 아니라면 “REJECTED”를 출력합니다.

B. 버섯 농장

#Graph_Traversal, #BFS, #DFS
출제진 의도- Normal

✅ 출제자: xingxing2001
✅ 제출 20번, 정답 8팀 (정답률 40.0%)
✅ 처음 푼 팀: KANKAN, 17분

B. 버섯 농장

- ✓ 버섯 농장의 상태를 2차원 배열로 입력 받습니다.
- ✓ 반복문을 통해 입력된 배열에서 버섯을 심을 수 있는 칸을 찾고, 해당 위치에서 BFS & DFS 를 진행하여 연결된 칸의 개수를 셉니다.
- ✓ 탐색 후 버섯이 자랄 수 있는 연결된 칸의 개수를 버섯이 자랄 수 있는 최대 값 K로 나눈 후 반올림을 합니다.

B. 버섯 농장

- ✓ 버섯 포자를 하나도 사용하지 않았다면 버섯 농사가 불가능하다고 판단합니다.
- ✓ 버섯 포자를 사용했다면, 심어야 하는 버섯 포자의 개수를 계속해서 구한 뒤 가지고 있는 버섯 포자 개수 M과 비교합니다.
- ✓ 만약, 현재 가지고 있는 버섯 포자의 개수가 심어야 하는 버섯 포자의 개수보다 작다면 버섯 농사는 불가능합니다.
- ✓ 반대로 현재 가지고 있는 버섯 포자의 개수가 심어야 하는 버섯 포자의 개수보다 크거나 같다면 버섯 농사는 가능하므로, POSSIBLE 출력 후 현재 버섯 포자의 개수를 출력하면 됩니다.

C. 연산자 파티

#Ad_Hoc
출제진 의도- Normal

✅ 출제자: wapas
✅ 제출 15번, 정답 0팀 (정답률 0.0%)

C. 연산자 파티

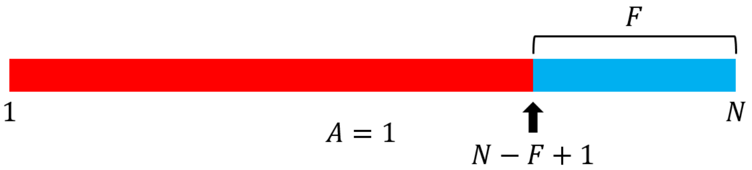
- ✓ 현재 변수 X에 담겨진 값을 x 라고 가정합니다.
- ✓ i 에 대해 연산을 진행했을 때, 비트 길이가 얼마나 커질 수 있을 지 확인합니다.

C. 연산자 파티

- ✓ 더하기 연산을 진행했을 때의 최대 비트 길이는 $\lfloor \log_2(\mathrm{x}+\mathrm{i}) \rfloor + 1$입니다.
- ✓ 나머지 연산을 진행했을 때의 최대 비트 길이는 $\lfloor \log_2(\mathrm{x}) \rfloor + 1$ 입니다.
- ✓ Bitwise AND 연산을 진행했을 때의 최대 비트 길이는 $\lfloor \log_2(\mathrm{x}) \rfloor + 1$ 입니다.
- ✓ Bitwise XOR 연산을 진행했을 때의 최대 비트 길이는 $\lfloor \log_2(\mathrm{x}) \rfloor + 1$ 입니다.
- ✓ Bitwise OR 연산을 진행했을 때의 최대 비트 길이는 $\lfloor \log_2(\mathrm{x}) \rfloor + 1$ 입니다.
- ✓ Bitwise R-Shift 연산을 진행했을 때의 최대 비트 길이는

  $\max(0, \lfloor \log_2(\mathrm{x}) \rfloor - i) + 1$ 입니다.

✓ X의 비트 길이를 최대로 늘린다고 가정한다면, 더하기 연산을 최대한 많이 시행했을 때의 경우는 $A = 1$이고 N이 F의 배수이면서 i가 $N - F + 1$부터 시작해서 N까지 증가할 때 입니다.



✓ Bitwise R-Shift 연산을 수행하기 전 X의 비트 길이는 $\left\lfloor \log_2\left(\frac{F(2N-F+1)}{2}\right) \right\rfloor + 1$ 입니다.

- ✓ 그런데 $\left\lfloor \log_2(\frac{F(2N-F+1)}{2}) \right\rfloor + 1 < N$ 이므로 N, F가 어떤 값이 되든 Bitwise R-Shift 연산을 진행한 직후에는 0 이 됩니다.

  $A > 1$ 인 경우에 대해서도 마찬가지입니다.
- ✓ 따라서 Bitwise R-Shift 연산을 진행하면 항상 X 는 0 이 됩니다.

$$if\ A = 1$$

$$F = 4$$

i	X	X binary
1	1	001
2	2	010
3	3	011
4	$4 \rightarrow 0$	$100 \rightarrow 000$

- ✓ 변수 X는 F를 주기로 값이 0이 되므로, 이 주기를 이용하여 계속 F만큼 계산을 건너뛸 수 있습니다.
- ✓ 따라서 $O(N)$이 아닌 $O(F)$ 시간복잡도로 문제를 풀 수 있습니다.

D. 수열 재배열

#Case_Work
출제진 의도- Normal

✅ 출제자: saywoo
✅ 제출 12번, 정답 1팀 (정답률 8.3%)
✅ 처음 푼 팀: KANKAN, 43분

D. 수열 재배열

- ✓ 연속된 K개의 수를 오름차순으로 재배열할 수 있으므로 답은 항상 K 이상이 됩니다.
- ✓ K개의 수의 구간을 재배열하여 K보다 큰 길이의 증가하는 부분 수열을 만들기 위해선, 구간의 앞 또는 뒤에 있는 수들을 포함하여 증가하는 부분 수열을 만들어야 합니다.

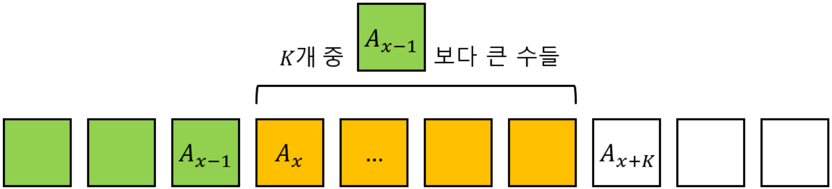
D. 수열 재배열

- ✓ 수열 A에서 임의의 연속한 K개의 수를 각각 $A_x, A_{x+1}, \ldots, A_{x+k-1}$ 라 정의합니다.
- ✓ S_i := A_i 로 시작하는 연속된 증가하는 부분 수열의 최대 길이
- ✓ E_i := A_i 로 끝나는 연속된 증가하는 부분 수열의 최대 길이

i	1	2	3	4	5	6
A_i	1	2	3	2	4	1
S_i	3	2	1	2	1	1
E_i	1	2	3	1	2	1

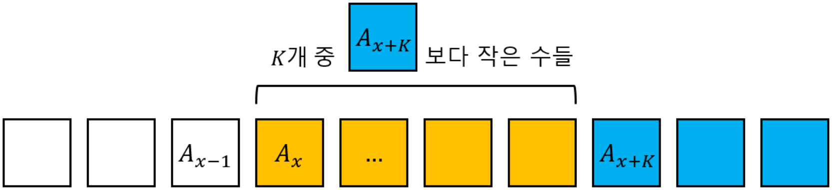
D. 수열 재배열

- ✓ 구간 앞에 있는 수들을 포함하여 증가하는 부분 수열을 만드는 경우
  - ✓ E_{x-1} + (구간의 수 중 A_{x-1} 보다 큰 수의 개수)가 최대 길이가 됩니다.



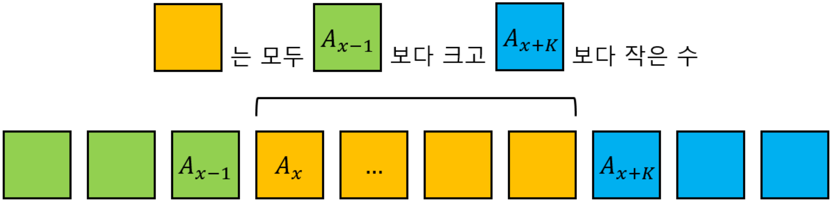
D. 수열 재배열

- ✓ 구간 뒤에 있는 수들을 포함하여 증가하는 부분 수열을 만드는 경우
  - ✓ (구간의 수 중 A_{x+k} 보다 작은 수의 개수) + S_{x+k} 가 최대 길이가 됩니다.



D. 수열 재배열

- ✓ A_{x-1} 보다 구간의 모든 수가 크고, A_{x+k} 보다 구간의 모든 수가 작을 때, 구간의 수를 오름차순으로 재배열한 경우
    - ✓ $E_{x-1} + K + S_{x+k}$가 최대 길이가 됩니다.



D. 수열 재배열

$$if\ K = 2\ \&\ A_4 \sim A_5$$

i	1	2	3	4	5	6
A_i	1	6	3	2	4	1
S_i	2	1	1	2	1	1
E_i	1	2	1	1	2	1

Case 1. $E_3 + 1 = 2$

Case 2: $S_6 + 0 = 1$

Case 3: -

D. 수열 재배열

- ✓ 수열 A에서 모든 연속한 K개의 수의 구간에 대해 앞에서 서술한 세 가지 경우의 값 중 최대 길이가 답이 됩니다.
- ✓ 임의의 K개의 수의 구간을 재배열할 때 만들 수 있는 증가하는 부분 수열의 최대 길이를 $O(N)$에 구할 수 있습니다.
- ✓ 총 시간복잡도는 O(N^2)가 됩니다.

E. 시프트 연산

#Greedy, #Constructive
출제진 의도- Hard

✅ 출제자: njw1204
✅ 제출 8번, 정답 1팀 (정답률 12.5%)
✅ 처음 푼 팀: KANKAN, 63분

- ✓ 최적의 방법은 LL...L, RR...R과 같이 한 방향으로만 계속 시프트하는 방법 또는 LL...LRR...R, RR...RLL...L과 같이 중간에 시프트 방향이 1회 달라지는 방법 중 하나입니다.
- ✓ 방향 전환이 2회 이상 이뤄지면 불필요한 이동이 포함되기 때문에 최적의 방법이 될 수 없습니다.
- ✓ 방향 전환을 1회 이하로 하는 방법들만 체크해보고 가장 연산 횟수가 적은 방법을 고르면 됩니다.

- ✓ 우선 주어진 수열에서 1이 등장하는 위치를 오름차순으로 나열한 길이 k의 수열 $X_1, X_2, \ldots, X_k$를 만듭니다.
- ✓ LL...L과 같이 L-Shift만 적용하는 방법을 쓴다면 최소 X_k회, RR...R과 같이 R-Shift만 적용하는 방법을 쓴다면 최소 $(N+1-X_1)$회 연산이 필요합니다.
- ✓ LL...LRR...R과 같이 L-Shift를 일정 횟수 적용 후 R-Shift를 일정 횟수 적용하는 방법을 쓴다면, 임의의 $i(1 \le i \le k-1)$에 대하여 L-Shift를 X_i 회 적용 후 R-Shift를 $X_i + (N+1-X_{i+1})$회 적용하면 수열을 모두 0으로 만들 수 있습니다. 최소 $\min(2X_i - X_{i+1} + N + 1)$회 연산이 필요합니다.
- ✓ RR...RLL...L도 같은 방법으로 계산할 수 있습니다.

i	**1**	**2**	**3**	**4**	**5**
A_i	0	1	1	0	0

$$X = \{2, 3\}$$

$$LL..L \to 3$$
$$RR..R \to 5 - 2 + 1$$

i	1	2	3	4	5
A_i	0	1	1	0	0

$$X = \{2, 3\}$$

$$LL..RR \rightarrow \min(2+5, -) = 7$$
$$RR..LL \rightarrow \min(-, 3+5) = 8$$

- ✓ 이렇게 확인한 방법들 중 가장 연산 횟수가 적은 방법을 출력하면 됩니다.
- ✓ $O(N)$개의 방법을 확인해야 되고, 각 방법의 연산 횟수를 계산하는데 $O(1)$의 시간복잡도가 걸립니다.
- ✓ 총 시간복잡도는 $O(N)$입니다.

F. 견제 미로찾기

#DP, #Game_Theory, #Number_Theory
출제진 의도- Hard

✅ 출제자: dkvltmxhf
✅ 제출 9번, 정답 1팀 (정답률 11.1%)
✅ 처음 푼 팀: KANKAN, 156분

- ✓ 건이와 준성이에게 최선의 전략이란, 필승으로 가는 수를 뜻합니다.
- ✓ 예를 들어, $k = 1$ 이고 미로에 벽이 없다고 가정했을 때, 자신의 차례에 말을 (N, N)에 위치시키게 되면 승리합니다.
- ✓ 반대로 자신의 차례에 말을 $(N - 1, N)$ 혹은 $(N, N - 1)$에 놓게 되면 패배합니다.
- ✓ 자신의 이전 차례에 말을 $(N - 2, N), (N - 1, N - 1), (N, N - 2)$에 놓게 되면 승리합니다.

- ✓ 설명에 앞서, (N, N)과 같이 자신이 말을 놓으면 승리할 수 있는 칸을 ‘승리 상태’, 패배하는 칸을 ‘패배 상태’ 라 정의합니다.
- ✓ 임의의 칸 (i, j)를 살펴보면, $(i\ +\ 1, j)$ 혹은 $(i, j + 1)$ 중 하나라도 승리 상태가 있게 되면 (i, j)가 패배 상태가 됩니다.
  - ✓ 내가 (i, j)로 이동하면 상대는 승리 상태의 지점으로 이동하므로 패배합니다.
- ✓ 따라서 확실한 승리상태 (N, N) 부터 시작하여 미로의 모든 칸을 승리 상태와 패배 상태로 구분 지을 수 있습니다.

- ✓ 이제 벽을 놓고, K를 확장 시켜 생각해봅시다.
- ✓ 미로 위 임의의 칸 (i, j)와 현재 K를 합한 상태인 (i, j, k)를 살펴봐야합니다.
- ✓ 상태 (i, j, k)로부터 갈 수 있는 상태는 크게 3가지로 나뉩니다.
  1. $(i + 1, j, k), (i + 2, j, k), \ldots, (i + k, j, k)$
  2. $(i, j + 1, k), (i, j + 2, k), \ldots, (i, j + k, k)$
  3. (i, j, l) - 이때 l은 k가 아닌 k의 약수들
- ✓ 물론 1과 2의 경우에는 중간에 장애물이 없어야 합니다.
- ✓ 1, 2, 3의 모든 케이스 중에 단 하나라도 승리 상태가 있으면 (i, j, k)는 패배 상태가 됩니다.

F. 견제 미로찾기

- ✓ 따라서 확실한 승리 상태인 $(N, N, 1)$ 부터 시작하여 모든 칸을 승리 상태와 패배 상태로 구분 지을 수 있습니다.
- ✓ 이후 $(1, 1, k)$이 승리 상태면 1, 패배 상태면 0을 출력하면 됩니다.
- ✓ d를 K의 약수의 개수라고 정의할 때, 시간복잡도는 모든 칸의 모든 약수를 살펴 봐야 하므로 $O(N^3 d)$입니다.

G. 월드컵 조별리그

#Binary_Search, #Implementation
출제진 의도- Hard

✅ 출제자: sldow4624, wapas
✅ 제출 2번, 정답 0팀 (정답률 0.0%)

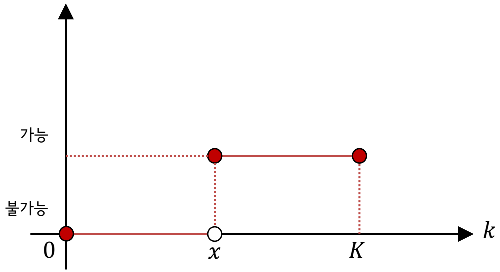
- ✓ ? 에 들어가는 값을 k 라고 정의합니다.
- ✓ k에 0부터 K까지 값을 대입하며 진출 여부를 확인합니다.
- ✓ 진출 자격을 획득할 수 있는 경우
  - ✓ 만족하는 k 값 중 최솟값이 K 이하인 경우
- ✓ 진출 자격을 획득할 수 없는 경우
  - ✓ 만족하는 k 값 중 최솟값이 K 보다 큰 경우
  - ✓ k 값과 상관없이 진출 자격을 획득 할 수 없는 경우

- ✓ k 값을 0부터 K까지 1씩 증가시키면서 브루트포스를 진행하면 O(K) 안에 풀 수 있습니다.
- ✓ 하지만 위 방법대로 풀면 시간초과를 받으므로 최적화가 필요합니다.
- ✓ 진출 자격을 획득할 수 있는 k 의 최솟값이 존재하고 그 값을 x라고 한다면 다음과 같은 특징을 관찰 할 수 있습니다.
  - ✓ $k < x$: 진출 자격을 획득할 수 없습니다.
  - ✓ $k \geq x$: 진출 자격을 획득할 수 있습니다.

G. 월드컵 조별리그

✓ 따라서 변수 k 에 대해서 결과가 x 값을 기준으로 진출 자격 획득 불가능, 진출 자격 획득 가능 영역이 나뉩니다.

✓ 그러므로 k 값에 대한 이분 탐색으로 $O(\log K)$ 안에 풀 수 있습니다.



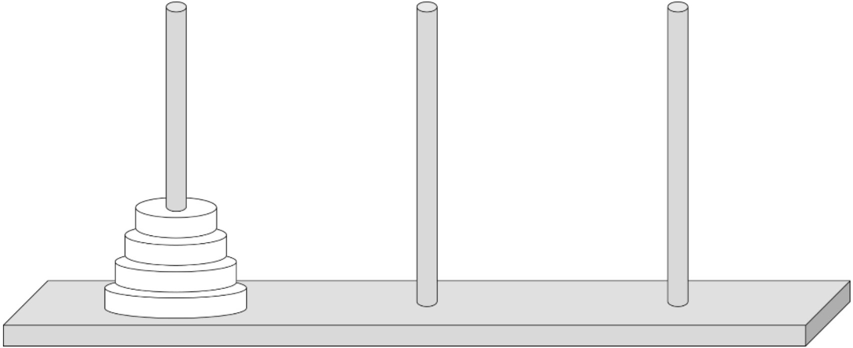
H. 어려운 하노이 탑

#Combinatorics, #Exponentiation_by_Squaring
출제진 의도- Challenging

✅ 출제자: wapas
✅ 제출 33번, 정답 0팀 (정답률 0.0%)

- ✓ 문제에서 입력으로 주는 N, M을 풀이에서는 각각 n, m으로 표현합니다.
- ✓ 1번 막대기를 $S(Source)$, 2번 막대기를 $A(Auxiliary)$,
  3번 막대기를 $T(Destination)$라고 표현합니다.
- ✓ 풀이에서 언급하는 i는 $1 \leq \ i \ \leq \ N$를 만족하는 양의 정수를 의미합니다.

✓ 일반 하노이 탑 $(m = 1)$



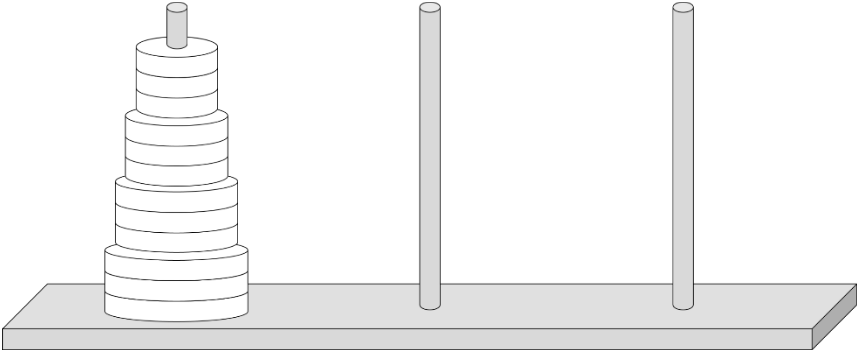
일반 하노이 탑 $(m = 1)$

- ✓ 일반 하노이 탑 문제에서 S에서 T로 옮기기 위해 원판을 이동하는 최소 횟수는 다음과 같은 공식으로 알려져 있습니다.
- ✓ $P(n) :=$ S에 크기가 i인 원판이 1개 존재할 때, S에 있는 원판을 쌓인 순서 그대로 모두 T로 옮기기 위해 소요되는 최소 횟수
- ✓ 점화식 일반항을 구하면 $P(n) = 2^n - 1$ 입니다.

$$\begin{cases} P(1) = 1 \\ P(n) = P(n-1) + 1 + P(n-1)\ for\ n \geq 2 \end{cases}$$

$$\therefore P(n) = 2^n - 1$$

✓ 원판에 번호가 없는 하노이 탑



원판에 번호가 없는 하노이 탑

- ✓ 원판이 번호가 없는 경우의 상황에서 S에서 T로 옮기기 위해 원판을 이동하는 최소 횟수를 구해봅니다.
- ✓ $Q(n) := $ S에 크기가 i인 원판이 m개 존재할 때, S에 있는 원판을 쌓인 순서 상관없이 모두 T로 옮기기 위해 소요되는 최소 횟수
- ✓ 점화식 일반항을 구하면 $Q(n) = m(2^n - 1)$ 입니다.

$$\begin{cases} Q(1) = m \\ Q(n) = Q(n-1) + m + Q(n-1)\ for\ n \geq 2 \end{cases}$$

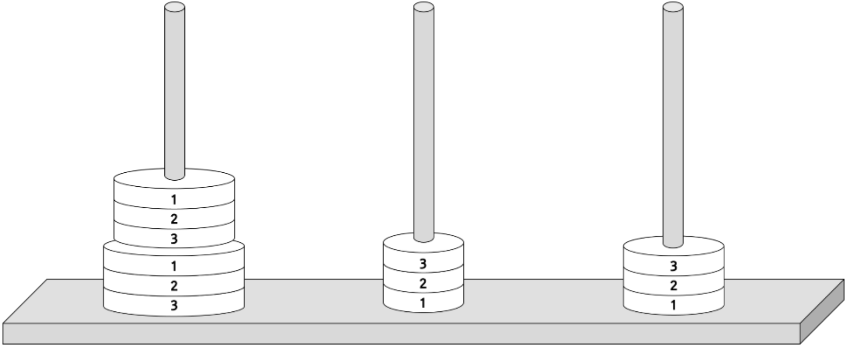
$$\therefore Q(n) = m(2^n - 1)$$

원판에 번호가 없는 하노이 탑

- ✓ 위의 점화식에 따르면 크기가 같은 원판끼리 항상 묶여서 이동한다는 것을 알 수 있습니다.
- ✓ 원판에 번호가 매겨져 있다고 가정하고 $Q(n)$의 방법으로 원판을 이동시켰을 때, T에 모든 원판이 쌓여 있는 최종 모습을 한 번 관찰해봅니다.

원판에 번호가 없는 하노이 탑

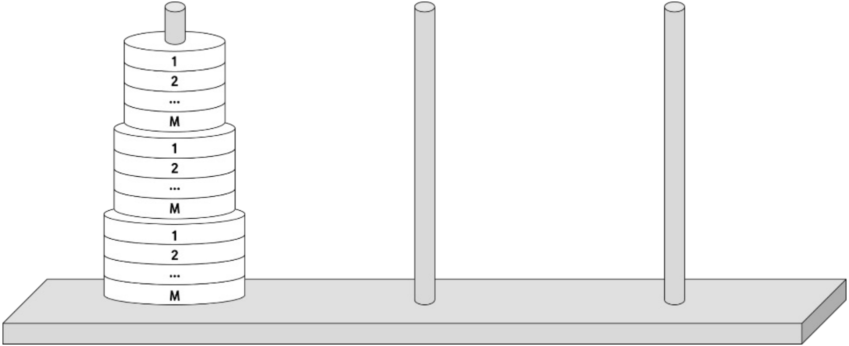
✓ T에 쌓인 원판 관찰



원판에 번호가 없는 하노이 탑

- ✓ 크기가 i인 원판 묶음이 다른 막대기로 1회 이동할 때마다 쌓이는 순서가 반전이 됩니다.
- ✓ 크기가 i인 원판이 오름차순으로 쌓여 있다면 다른 막대기로 1번 이동하면 내림차순 순서로, 내림차순으로 쌓여 있다면 오름차순 순서로 쌓입니다.
- ✓ 그러므로 크기가 i인 원판 묶음이 S에서 출발하여 막대기를 짝수번에 걸쳐 T에 도착했다면 S에 쌓인 순서 그대로, 홀수번에 걸쳐 T에 도착했다면 S에 쌓인 순서 반대로 쌓이게 됩니다.

✓ 원판에 번호가 있는 하노이 탑



원판에 번호가 있는 하노이 탑

- ✓ 문제 상황 상황에서 S에서 T로 옮기기 위해 원판을 이동하는 최소 횟수를 구해봅니다.
- ✓ $R(n) :=$ S에 크기가 i인 원판이 m개 존재할 때, S에 있는 원판을 쌓인 순서 그대로 모두 T로 옮기기 위해 소요되는 최소 횟수

원판에 번호가 있는 하노이 탑

✓ $m = 1$일 때 일반 하노이 탑 상황과 같으므로 점화식은 다음과 같습니다.

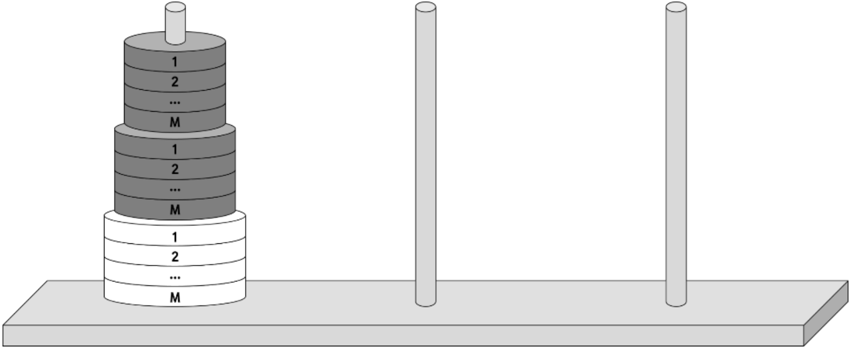
$$R(n) = P(n) \; if \; m \; = \; 1$$

✓ $m \geq 2, n = 1$일 때 번호가 없는 하노이 탑 상황과 같으므로 점화식은 다음과 같습니다.

$$R(1) = Q(1) \; if \; m \geq 2$$

원판에 번호가 있는 하노이 탑

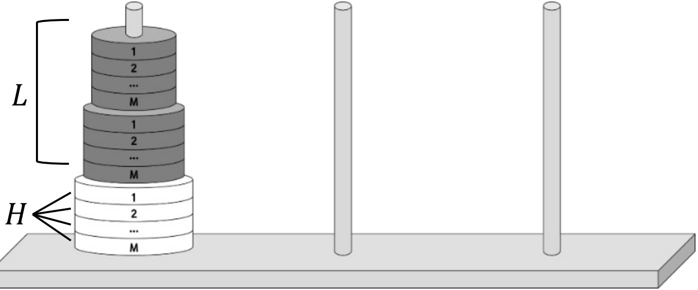
- ✓ $m \ge 2, n \ge 2$일 때
- ✓ 크기가 n인 원판의 묶음과 크기가 n이 아닌 원판의 묶음의 관계를 살펴봅니다.



원판에 번호가 있는 하노이 탑

✓ 크기가 가장 큰 원판들을 각각 H, 크기가 가장 크지 않은 원판들의 묶음을 L이라고 정의합니다.

✓ L을 다른 막대기로 옮기는데 필요한 이동 횟수는 쌓이는 순서를 따진다면 $R(n-1)$, 순서를 따지지 않는다면 $Q(n-1)$입니다.



원판에 번호가 있는 하노이 탑

- ✓ S에서 T로 옮길 때, L의 이동 횟수를 x, H의 이동 횟수를 y라 정의합니다. $(x > 0, y > 0)$
- ✓ 그러면 $R(n)$ 은 다음과 같습니다.
  - ✓ $R(n) = Q(n-1)\times x + y \quad if\ x\ is\ even$
  - ✓ $R(n) = Q(n-1)\times(x-1) + R(n-1) + y \quad if\ x\ is\ odd$
- ✓ x와 y는 둘 다 양의 정수이기 때문에 x, y 모두 최소화를 해야 $R(n)$의 값을 얻을 수 있습니다.

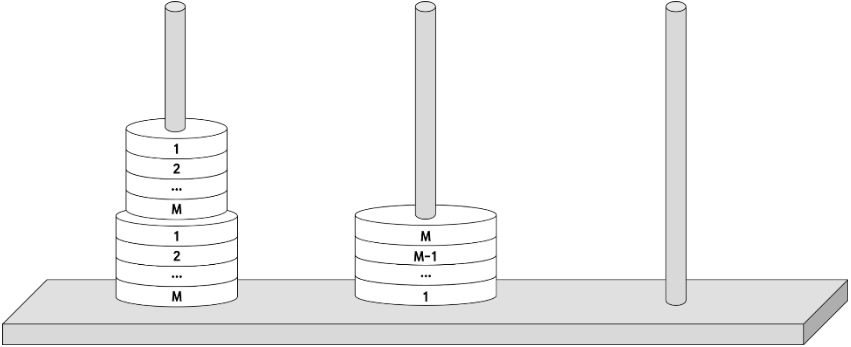
원판에 번호가 있는 하노이 탑

1) $x \leq 2$ 일 때

✓ 불가능 합니다.

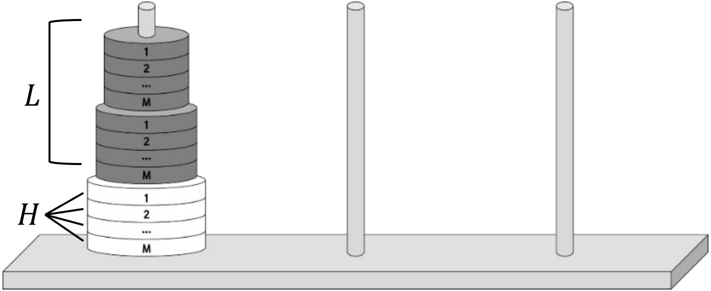
원판에 번호가 있는 하노이 탑

2) $x = 3$ 일 때



원판에 번호가 있는 하노이 탑

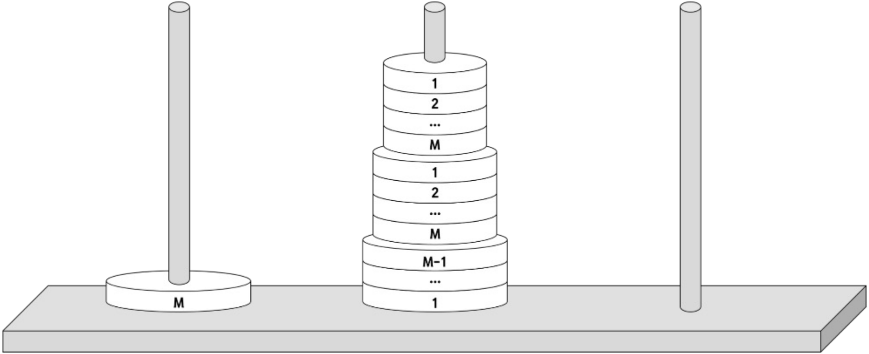
1. S에서 T로 L 이동
2. S에서 A로 H 이동
3. T에서 S로 L 이동
4. A에서 T로 H 이동
5. S에서 T로 L 이동



$$R_{x=3}(n) = Q(n-1) + m + Q(n-1) + m + R(n-1)$$
$$= Q(n-1) \times 2 + R(n-1) + 2m$$
$$R_{x=3}(n) = m2^{n+1} - 2m - 1$$

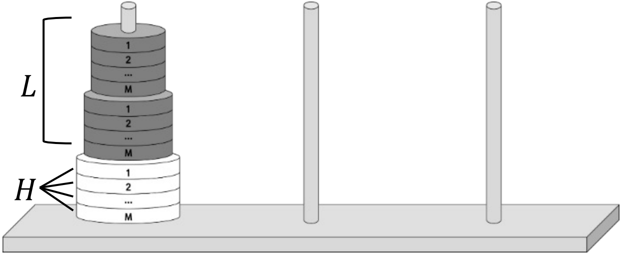
원판에 번호가 있는 하노이 탑

3) $x = 4$ 일 때



원판에 번호가 있는 하노이 탑

1. S에서 T로 L 이동
2. S에서 A로 H 중 상위 $m-1$개 이동
3. T에서 A로 L 이동
4. S에서 T로 1개의 H 이동
5. A에서 S로 L 이동
6. A에서 T로 $m-1$개의 H 이동
7. S에서 T로 L 이동



원판에 번호가 있는 하노이 탑

$$R_{x=4}(n) = Q(n-1) + (m-1) + Q(n-1) + 1 + Q(n-1) + (m-1) + Q(n-1)$$
$$= Q(n-1) \times 4 + (2m-1)$$

$$R_{x=4}(n) = m2^{n+1} - 2m - 1$$

원판에 번호가 있는 하노이 탑

4) $x \geq 5$ 이고 x가 홀수일 때

✓ 이 때 만족하는 최솟값 y는 $2m - 1$입니다.

✓ x가 홀수 일 때는 $x = 3$인 경우 이동횟수가 최소이므로 $x \geq 5$일 때는 항상 이동횟수가 더 많습니다.

5) $x \geq 5$ 이고 x가 짝수일 때

✓ 이 때 만족하는 최솟값 y는 $2m - 1$입니다.

✓ x가 짝수 일 때는 $x = 4$인 경우 이동횟수가 최소이므로 $x \geq 5$일 때는 항상 이동횟수가 더 많습니다.

H. 어려운 하노이 탑

- ✓ $R_{x=3}(n) = R_{x=4}(n) = R(n) = m2^{n+1} - 2m - 1$

$$\begin{cases} R(n) = 2^n - 1 \quad if\ m = 1 \\ R(n) = m2^{n+1} - 2m - 1 \quad if\ m \geq 2 \end{cases}$$

- ✓ 위 식을 통해 분할 정복을 이용한 거듭제곱으로 $O(\log n)$안에 풀 수 있습니다.
- ✓ 오버플로우가 발생하지 않게 연산에 주의합니다.

모두 수고하셨습니다!

중앙대학교 알고리즘 학회 ChAOS