

#1

①

A B C

a<sub>1</sub>  
a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>  
a<sub>5</sub>a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>  
a<sub>5</sub>a<sub>1</sub>a<sub>3</sub>  
a<sub>4</sub>  
a<sub>5</sub>a<sub>2</sub>a<sub>1</sub>a<sub>3</sub>  
a<sub>4</sub>  
a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>a<sub>4</sub>  
a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>4</sub>  
a<sub>5</sub>a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>  
a<sub>5</sub>a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>  
a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>5</sub>a<sub>4</sub>a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>2</sub>  
a<sub>5</sub>a<sub>1</sub>a<sub>4</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>2</sub>  
a<sub>5</sub>a<sub>4</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>2</sub>  
a<sub>5</sub>a<sub>3</sub>a<sub>4</sub>a<sub>2</sub>  
a<sub>5</sub>a<sub>3</sub>a<sub>4</sub>a<sub>1</sub>a<sub>5</sub>a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>a<sub>1</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>

②

A B C

a<sub>1</sub>  
a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>3</sub>  
a<sub>4</sub>a<sub>2</sub>a<sub>5</sub>a<sub>3</sub>  
a<sub>4</sub>a<sub>1</sub>a<sub>2</sub>a<sub>5</sub>a<sub>3</sub>a<sub>4</sub>a<sub>1</sub>a<sub>2</sub>a<sub>5</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>4</sub>a<sub>2</sub>a<sub>5</sub>a<sub>2</sub>a<sub>1</sub>  
a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>  
a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>  
a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>a<sub>2</sub>a<sub>1</sub>  
a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>2</sub>a<sub>4</sub>a<sub>5</sub>a<sub>3</sub>a<sub>1</sub>  
a<sub>2</sub>a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>  
a<sub>2</sub>a<sub>3</sub>a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>a<sub>5</sub>a<sub>1</sub>  
a<sub>2</sub>  
a<sub>3</sub>  
a<sub>4</sub>a<sub>5</sub>

#2

the pseudo code of Hanoi Tower in an iterative manner

hanoi\_tower(discs)

int pegs = discs + 1 ;

for (i=1 to pegs)

printf("Disk (i-1) moves to ith peg from 0 peg.\n");

for(j=discs-1 to 0)

printf("Disk (j-1) moves to (pegs-1)th peg from j peg.\n");

• the complexity

n개의 discs가 들어왔다고 가정하자.

- pegs 할당 ... (덧셈+할당) 2개

- 첫번째 for문 ... n-1

- 두 번째 for문 ... n

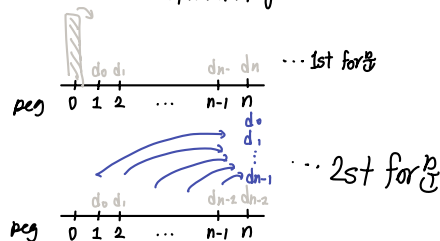
⇒ T(n) = O(n)

• how many bars?

가장 단순한 방법은 bar을 n+1 개만큼 만드는 것이다.

좌측에 있는 막대 0에서 가장 오른쪽에 있는 막대 n에 탑을 옮기기 위해서는 가장 위에 있는 디스크를 1번씩 밖에서부터 차례대로 이동시키는 것이다. 이 for문이 종료되면, n-1 막대에 있는 디스크부터 차례대로 n번씩 이동시킨다.

ex) n개의 dish, n+1개의 peg



#3

Explain the time complexity of the following equations

$$T(n) = T\left(\frac{n}{2}\right) + C$$

 $T\left(\frac{n}{2}\right)$ : 입력 크기가 절반으로 줄어든 부분 문제에 대한 수행시간.

C: 이외 기본 연산.

• 문제를 sub problem으로 나눌 때, 개수는 동일하게 하라.

problem의 크기는 절반으로 줄어든다.  $n=2^k$ 인 문제가 있다고 가정했을 때,sub problem으로 줄어들수록,  $n=2^k, 2^{k-1}, 2^{k-2}, \dots, 2^0$ 가 된다.

$$\therefore T(n) = O(\log_2 n)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + C$$

 $T(n) = T\left(\frac{n}{2}\right) + C$  와 달리  $T\left(\frac{n}{2}\right)$ 에 곱해진 2는

sub problem이 두 개가 되었음을 의미한다.

$$\therefore T(n) = O(n)$$

이 경우 재귀를 이용했음에도 Big O notation의 시간 복잡도가 동일하기 때문에

iteration을 이용하는게 더 낫다.