Homework

1. In Hanoi Tower, explain the process when 5 discs are used (as in p28 of ‘DS-Lec02-Recursion’).

2. Show the pseudo code of Hanoi Tower in an iterative manner, and explain what the complexity is, and how many bars are needed.
   (Note that the recursion of Hanoi Tower requires 3 bars only.)

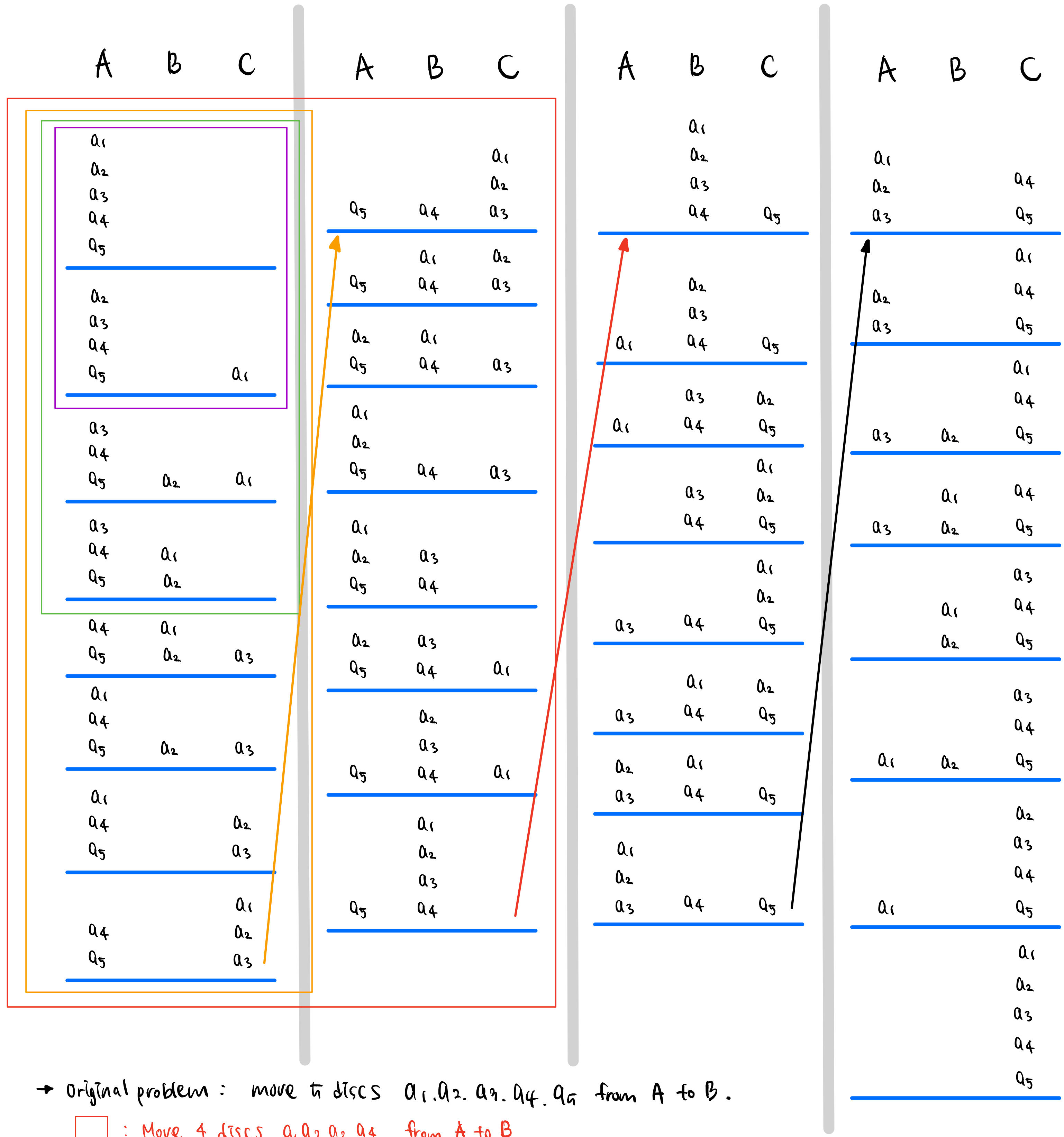
3. Explain the time complexity of the following equations.

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

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

1. In Hanoi Tower, explain the process when 5 discs are used (as in p28 of 'DS-Lec02-Recursion').

* Hierarchical Structure of Recursion (Devide-and-Conquer)



→ Original problem : move 5 discs $a_1 . a_2 . a_3 . a_4 . a_5$ from A to B.

- ☐ (red) : Move 4 discs a_1, a_2, a_3, a_4 from A to B
- ☐ (orange) : Move 3 discs a_1, a_2, a_3 from A to C
- ☐ (green) : Move 2 discs a_1, a_2 from A to B
- ☐ (purple) : Move 1 disc a_1 from A to C

2. Show the pseudo code of Hanoi Tower in an iterative manner, and explain what the complexity is, and how many bars are needed.
(Note that the recursion of Hanoi Tower requires 3 bars only.)

Idea

n개의 disc를 가진 하노이타워를 src 막대에서 dest 막대로 옮기기 위해서는 n번째 disc가 dest의 가장 아래쪽에 위치하도록 만들어줘야 한다. 따라서, 임시로 거쳐갈 막대 (temp)의 개수에 제한이 없다는 점을 이용하여 다음과 같은 절차로 문제를 해결한다.

① n번째 disc를 옮기기 위하여 1~n-1번째 disc를 차례로 temp로 옮긴다. 한번 옮길 때마다 새로운 temp 막대를 사용한다.
② n번째 disc를 src 에서 dest 막대로 옮긴다.
③ temp에 옮겨뒀던 1~n-1번째 disc를 역순으로 dest 막대로 옮긴다.

Pseudo Code

```
void hanoi_tower (int n) {
    struct bar temp[n-1];        // 총 n-1개의 temp 막대 필요
    if (n==1) { Move a disc from src to dest }
    else {
        for (i=0; i<n-1 ; i++) {
            move 1 disc from src to temp[i]
        }
        move a disc from src to dest
        for (i=n-2; i>=0; i--) {
            move 1 disc from temp[i] to dest
        }
    }
}
```

필요한 막대의 개수 src 1개, temp (n-1)개, dest 1개

⇒ 총 n+1개 필요

시간복잡도 disc를 옮기는 연산을 기준으로,

첫번째 for문 : 연산 n-1회 수행
n번째 disc를 src → dest : 연산 1회 수행
두번째 for문 : 연산 n-1회 수행

∴ $(n-1) + 1 + (n-1)$ ⇒ $O(n)$의 시간복잡도를 가진다.

3. Explain the time complexity of the following equations.

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

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

$$\cancel{T\left(\frac{n}{2}\right)} = \cancel{T\left(\frac{n}{4}\right)} + c$$

$$\cancel{T\left(\frac{n}{4}\right)} = \cancel{T\left(\frac{n}{8}\right)} + c$$

$$\vdots$$

$$\cancel{T\left(\frac{n}{2^{k-1}}\right)} = T\left(\frac{n}{2^k}\right) + c$$

$+$

$$T(n) = T\left(\frac{n}{2^k}\right) + c \cdot k$$

문제의 크기 $n = 2^k$ 형태라고 가정. (k는 양의정수.)

$\therefore k = \log_2 n$

$$\therefore T(n) = \cancel{T(1)} + \cancel{c} \cdot k$$

여기서 $T(1)$, c는 상수값이므로, $n \to \infty$, 즉 Asymptotic한 상황의 시간복잡도 계산에서는 무시할 수 있다.

$$\therefore T(n) \in \Theta(\log_2 n)$$

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

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

$$2 \cdot \cancel{T\left(\frac{n}{2}\right)} = \cancel{2^2 \cdot T\left(\frac{n}{2^2}\right)} + 2c$$

$$2^2 \cdot \cancel{T\left(\frac{n}{2^2}\right)} = \cancel{2^3 \cdot T\left(\frac{n}{2^3}\right)} + 2^2 c$$

$$\vdots$$

$$+ \quad 2^{k-1} \cdot \cancel{T\left(\frac{n}{2^{k-1}}\right)} = 2^k \cdot T\left(\frac{n}{2^k}\right) + 2^{k-1} \cdot c$$

$\leftarrow n = 2^k$

$k = \log_2 n$

$$T(n) = 2^k \cdot T\left(\frac{n}{2^k}\right) + \left(c + 2c + 2^2 c \cdots + 2^{k-1} \cdot c\right)$$

$$T(n) = 2^k \cdot T(1) + c \cdot \left(2^{k-1} - 1\right)$$

$$T(n) = n \cdot \cancel{T(1)} + \cancel{c} \cdot \left(\frac{1}{2} \cdot n - \cancel{1}\right)$$

여기서 T(1), C 의 값은 상수이므로 무시 가능하다.

$$\therefore T(n) \in \Theta(n)$$