

# Question 1

Sunday, January 20, 2019 12:44 AM

(a)

```
void Hanoi(int n) {
    if(!s->v->isEmpty() && n >= 1) {
        Hanoi(n-1);
        moveNext(s, a2);
        H1(n-1, a3, a2, a1);
        moveNext(a2, a3);
        if(!s->v->isEmpty())
            H2(n-1, a1, a2, a3);
        else
            Hanoi(n-1);
    } else if(s->v->isEmpty()) {
        if(n == 0)
            moveNext(a3, d);
        else if(n >= 1) {
            moveNext(a3, d);
            H2(n-1, a1, a2, a3);
            moveNext(a1, a2);
            H1(n-1, a3, a2, a1);
            moveNext(a2, a3);
            Hanoi(n-1);
        }
    }
    recursiveCalls++;
}
```

```
void H2(int n, Node * begin, Node * aux, Node * end) {
    if(n == 1)
        moveNext(begin, end);
    else if(n >= 2) {
        H2(n-1, begin, aux, end);
        moveNext(begin, aux);
        H1(n-1, end, aux, begin);
        moveNext(aux, end);
        H2(n-1, begin, aux, end);
    } recursiveCalls++;
}

void H1(int n, Node * begin, Node * aux, Node * end) {
    if(n == 1)
        moveNext(begin, end);
    else if(n >= 2) {
        H2(n-1, begin, end, aux);
        moveNext(begin, end);
        H2(n-1, aux, begin, end);
    } recursiveCalls++;
}
```

## Time Complexity

Where  $Hanoi(n)$  is  $T(n)$ ,

$$Hanoi(n) = 3 \cdot Hanoi(n-1) + 2 \cdot Hop(n-1) + f(n) \Rightarrow O(3^n)$$

Explanation

$3 \cdot Hanoi(n-1)$  line 1, 3, and 20 of *Hanoi* function

$$\Rightarrow O\left(\frac{n}{3-1}\right) \Rightarrow O(3^n)$$

$2 \cdot Hop(n-1)$  *H1* and *H2* call from both *stack is empty* & *stack is not empty*

$$\Rightarrow O\left(\frac{n}{2-1}\right) \Rightarrow O(2^n)$$

$f(n)$  = can either be 2 or 3-4 moves

where it's 2 moves, stack is not empty (line 4 and 6 show this)

where it's 3-4 moves, stack is empty (line 13, 15, 17, and 19)

$$\Rightarrow O(1)$$

$3 \cdot Hanoi(n-1)$  in this case has the largest long-term growth

$$\therefore Hanoi(n) = O(3^n)$$

## Space Complexity

Where *stack size* is the number of disks  $n$

# of array stacks  $\cdot$  stack size + atomic moves per recursive call  $\cdot$  recursive calls

$$= 5 \cdot n + \underset{\text{stack is not empty}}{2 \cdot [3(n-1)]} + \underset{\text{stack is empty}}{4 \cdot [3(n-1)]}$$

$$= O(n) \quad \text{linear growth}$$

(b) On next page

Assignments Page 2