

Time Complexity – Competitive Practice Sheet

1. Find the time complexity of the func1 function in the program show in program1.c as follows:

```
#include <stdio.h>

void func1(int array[], int length)
{
    int sum = 0;
    int product = 1;
    for (int i = 0; i < length; i++)
    {
        sum += array[i];
    }

    for (int i = 0; i < length; i++)
    {
        product *= array[i];
    }
}

int main()
{
    int arr[] = {3, 5, 66};
    func1(arr, 3);
    return 0;
}
```

$$\begin{aligned} T_n &= f_1 + f_2 + f_3 \\ &= k_1 + k_2 n + k_3 n \\ &\Rightarrow (k_2 + k_3) n \\ &= k_4 n \rightarrow O(n) \\ &O(\text{length}) \end{aligned}$$

2. Find the time complexity of the func function in the program from program2.c as follows:

```
void func(int n)
{
    int sum = 0;
    int product = 1;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("%d , %d\n", i, j);
        }
    }
}
```

$$\begin{aligned} &\rightarrow [n + n + n + \dots + (n-1)n] k_2 \\ &n k_2 (1 + 1 + \dots + 1) = k_2 n^2 \\ &\quad \quad \quad \underbrace{\hspace{1cm}}_{n \text{ times}} \end{aligned}$$

3. Consider the recursive algorithm ~~above~~ <sup>below</sup> where the random(int n) spends one unit of time to return a random integer which is evenly distributed within the range [0,n] ~~0,1~~ <sup>0,n-1</sup>. If the average processing time is T(n), what is the value of T(6)?

(-)

random(6) → [0, 6]

$O(n^2)$

[0,5]

```
int function(int n)
{
    int i;  $\rightarrow k_1 = 0$ 
    if (n <= 0)
    {
        return 0;
    }
    else
    {
        i = random(n - 1);  $\rightarrow 1$ 
        printf("this\n");
        return function(i) + function(n - 1 - i);
    }
}
```

4. Which of the following are equivalent to  $O(N)$ ? Why?  $(k_1 + k_2) \rightarrow O(n)$

- a)  $O(N + P)$ , where  $P < N/9$   $\rightarrow O(n)$
- b)  $O(N + P) \rightarrow O(n)$
- c)  $O(N + \log N)$
- d)  $O(N + M^2)$

5. The following simple code sums the values of all the nodes in a balanced binary search tree. What is its runtime?  $(n \text{ is the no of nodes})$

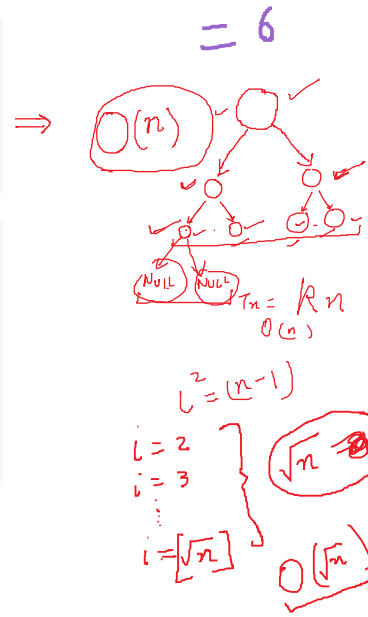
```
int sum(Node node)
{
    if (node == NULL)
    {
        return 0;
    }
    return sum(node.left) + node.value + sum(node.right);
}
```

6. Find the complexity of the following code which tests whether a given number is prime or not?

```
int isPrime(int n){
    if (n == 1){
        return 0;
    }

    for (int i = 2; i * i < n; i++) {
        if (n % i == 0)
            return 0;
    }

    return 1;
}
```



7. What is the time complexity of the following snippet of code?

```
int isPrime(int n){
    for (int i = 2; i * i < 10000; i++) {
        if (n % i == 0)
            return 0;
    }

    return 1;
}
```

$k_1$   $O(1)$   
 $T_n = k_1$





