Questions

02 August 2020 19:40

Time Complexity – Competitive Practice Sheet

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

```c
#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;
}
```

$f_1 = k_1$

$f_2 = k_2 n$

$f_3 = k_3 n$

$\longrightarrow T_n = f_1 + f_2 + f_3$

$= k_1 + k_2 n + k_3 n$

$\Rightarrow (k_2 + k_3) n$

$= k_4 n \rightarrow O(n)$

$O(\text{length})$

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

```c
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);
        }
    }

}
```

k_1

0 to n

0 to n $\rightarrow k_2$

$\longrightarrow [n + n + n + \cdots + (n-1)n] k_2$

$nk_2 (\underbrace{1 + 1 + \cdots 1}_{n \text{ times}}) = k_2 n^2$

3. Consider the recursive algorithm ~~above~~ below, 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,n~~], If the average processing time is T(n), what is the value of T(6)?

(–)

random(6) → [0, 6]

$O(n^2)$

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

4. Which of the following are equivalent to O(N)? Why?
   a) O(N + P), where P < N/9
   b) O(N-1)
   c) O(N + log N)
   d) O(N + M²)

5. The following simple code sums the values of all the nodes in a balanced binary search tree. What is its runtime?

```c
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 give number is prime or not?

```c
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?

```c
int isPrime(int n){

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

    return 1;
}
isPrime(4);
```