Time Complexity ▾ | Practice Set | +

Questions

02 August 2020 19:40

Time Complexity – Competitive Practice Sheet

1. Fine 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;
}
```

$f_1 = k_1$

$f_2 = k_2 n$

$f_3 = k_3 n$

$\longrightarrow T_n = f_1 + f_2 + f_3$

$= k_1 + k_2 n + k_3 n$

$\Rightarrow (k_2 + k_3) n$

$= k_4 n \rightarrow O(n)$

2. Fine 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);
```

Time Complexity ▾ | Practice Set | +

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);
```

Time Complexity ▾ | Practice Set | +

```
    }

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

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

$f_3 = k_3 n$

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);
        }
    }

}
```

k_1 ; 0 to n ; 0 to n ; k_2

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

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

$O(n^2)$

3. Consider the recursive algorithm above, 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(5)?

```
int function(int n)
{
    int i;

    if (n <= 0)
    {
        return 0;
    }
```

$= (k_1 + k_2 \cdots$

$\Rightarrow (k_2 + k_3) n$

$= k_4 n \rightarrow O(n)$

$O(\text{length})$

}

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)? (—)

$n R_2 \{ 1 + 1 + \cdots$ (n times)

$O(n^2)$

```
int function(int n)
{
    int i; ] → k1=0

    if (n <= 0)
    {
        return 0;
    }
    else
    {
        i = random(n - 1); → 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(9N-k)

Time Complexity ▾ | Practice Set | +

random integer which is evenly distributed within the range [0,n]. If the average processing time is T(n), what is the value of T(6)? (—)

random(6) → [0, 6]

[0, 5]

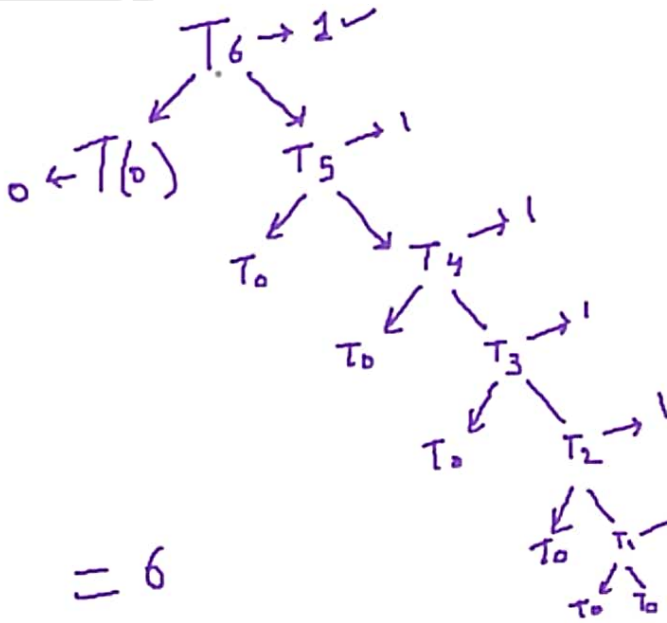
```
int function(int n)
{
    int i; → k = 0
    if (n <= 0)
    {
        return 0;
    }
    else
    {
        i = random(n - 1); → 1
        printf("this\n");
        return function(i) + function(n - 1 - i);
    }
}
```

4. Which of the following are equivalent to O(N)? Why?
   a) O(N + P), where P < N/9
   b) O(9N k)
   c) O(N + 8log 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?

```
int sum(Node node)
{
    if (node == NULL)
    {
        return 0;
    }
```

$O(n^2)$



= 6

Time Complexity ▾ Practice Set +

```
        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 → O(N)

b) O(9N-K) → O(N)

c) O(N + 8log N) → O(N)

d) O(N + M²) →

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

```
int sum(Node node)
{
    if (node == NULL)
    {
```

Time Complexity ▾ Practice Set +

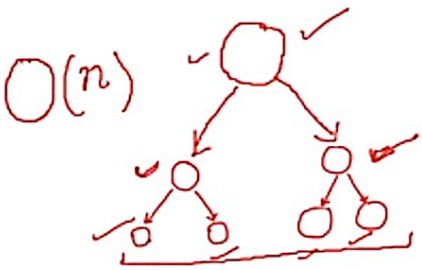
4. Which of the following are equivalent to O(N)? Why?
   a) O(N + P), where P < N/9 → O(N)
   b) O(N-K) → O(N)
   c) O(N + 8log N) → O(N)
   d) O(N + M²) →

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

(n is the no of nodes)

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

⇒ O(n)



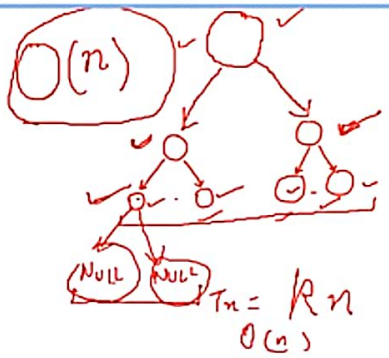
= 6

6. Find the complexity of the following code which tests whether a give 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 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?

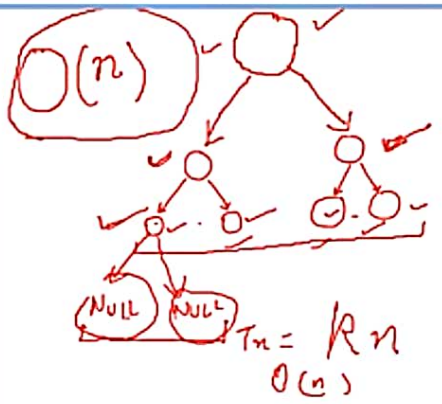
```
int isPrime(int n){
    if (n == 1){
        return 0;
    }

    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?

```
    {
        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?

```
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;
}
```

$K_1 + K_2(\ )$

$i^2 = (n-1)$

$i = 2$
$i = 3$
$\vdots$
$i = \lceil \sqrt{n} \rceil$

$\sqrt{n}$

$O(\sqrt{n})$

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

$i = [\sqrt{n}]$ $O(\sqrt{n})$

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;
}
isPrime(n);
```

k_1

$T_n = k_1 \rightarrow O(1)$