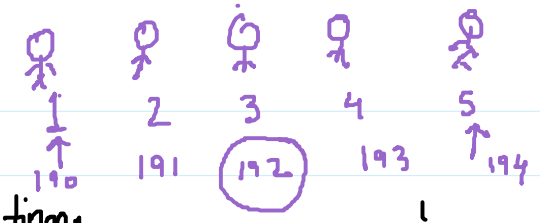
ADTs

Delhi



ADTs = Set of Values + Set of operations

int → 9, 10, 12

9 + 12 = 21

operator

Details abstracted

```
 1 2
+ 9
---
 2 1
```

myArray →

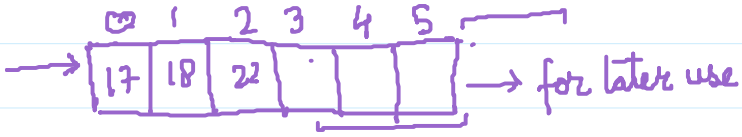
representation

- total_size → 6
- used_size → 3
- base address

+

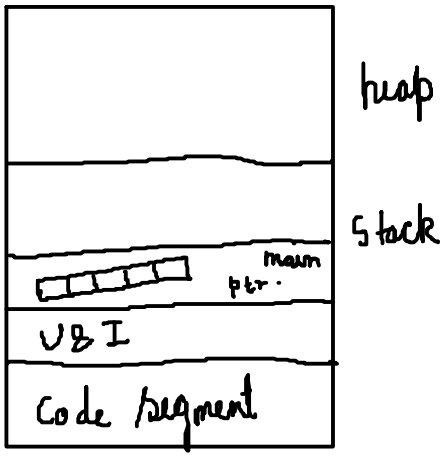
operations

- max()
- get(i)
- set(i, num)
- add(arr)

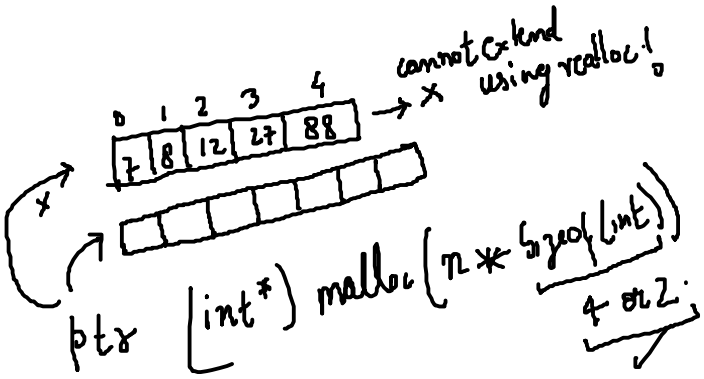


Stack/Heap

05 August 2020 12:32



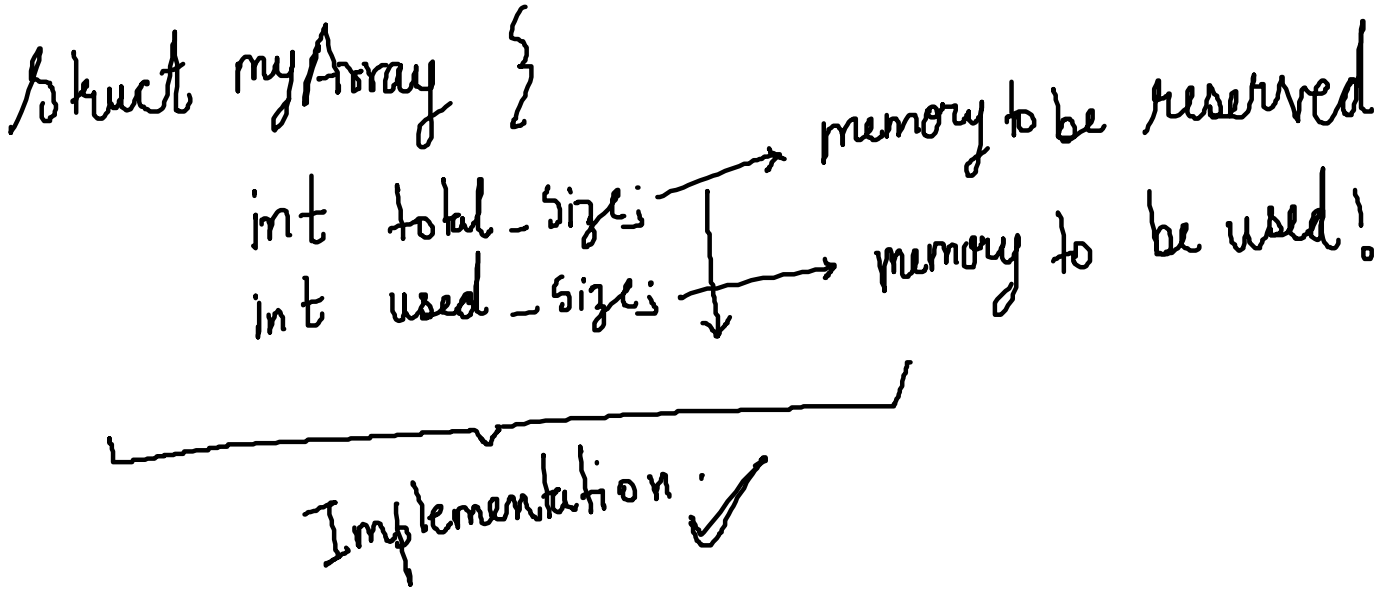
Memory



Code

05 August 2020 12:38

C language Code



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;
}
```

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);
        }
    }

}
```

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

```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-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?

```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?

```
int isPrime(int n){

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

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