Course: COEN279/AMTH377

Lecturer: Yuan Wang

Department: Computer Engineering

Data/Time: 2021/04/28 7:10am - 8am

Name: Mansi Jainendra Tandel

Student id: W1606463

1. If c_2 and n_0 exist such that $6n^3 \leq c_2$ for all $n \geq n_0$.
   Then $n \leq c_2/6$, is a contradiction.
   Since c_2 is constant, it cannot possibly hold for arbitrary large n
   Hence, it is proved that, $6n^3$ is not equal to $\Theta(n^2)$

2. MyFunction(n) { running time

   i, j, k, count = 0 1

   for i = n/2; i<=n; i = i+1 n/2

   for j = 1; j<=n; j = 2*j (n/2-1)log n

   for k =1; k<=n; k = k*2 (n/2-1)(log n-1)log n

   count = count + 1 (n/2-1))(log n-1)(log n-1)

}

Time complexity=1+(n/2)+ (n/2-1)log n+(n/2-1)(log n)log n+ n/2-1))(log n)(log n)=(n/2)(log $_2$ n)^2

=n/2(n)

=O(n^2 logn)

3. $T(n) = 9T(n/3) + T(n^2)$

We have a = 9, b = 3, $f(n) = \Theta(n^2)$

$n^{\log_b a} = n^{\log_3 9} = n^2$

, so $f(n) = \Theta(n^{\log_b a})$.

This is Case 2 of the Master Theorem. Thus,

$T(n) = \Theta(n^2 \log n)$.

4.

```
// Function to find the winner present in a given array
public static int findwinner (int[] A)
{
   // `w` stores the winner
   int w = -1;                                          c1

   // initialize counter `i` with 0
   int i = 0;                                           c2

   // do for each element `A[j]` in the array
   for (int j = 0; j < A.length; j++)                   n+1
   {
      // if counter `i` becomes 0
      if (i == 0)                                       c3
      {
         // set the current candidate to `A[j]`
         w = A[j];                                      c4*(n)

         // reset counter to 1
         i = 1;                                         c5
      }

      // increments the counter if `A[j]` is a current candidate
      else if (w == A[j]) {
         i++;                                           c6
      }
      // decrements the counter if `A[j]` is a current candidate
      else {
         i--;                                           c7
      }
   }

   return w;                                            c8
```

```
    }

    public static void main (String[] args)
    {
        // If Winning person is present
//here, b represents array of candidates
        int[] b= {//array of candidates}                    c9

        System.out.println("The winning person is " + findwinner(b));   c10
    }
```

Time complexity= c_1 +c_2 +n+1+ c_3 +c_4*(n)+c_5+c_6+ c_7+c_8+ c_9+c_{10}
So, the time complexity T(n) = O(n).

Here, the algorithm is a linear algorithm so the time complexity will be O(n.)