Practise problems on Time complexity of an algorithm

1. Analyse the number of instructions executed in the following recursive algorithm for computing nth Fibonacci numbers as a function of n

```
public static int fib(int n)
{
     if(n==0) return 1;
     else if(n==1) return 1;
          else return(fib(n-1) + fib(n-2));

}
public static void main(String args[])
{
     int n = Integer.parseInt(args[0]);
     System.out.println(fib(n));
}
```

   Answer : The instructions executed by the above algorithm is c times the value of nth fibonacci number. For your information, the value of nth fibonacci number is exponential in n.
   (**Hint :** use recurrence or use recursion tree as used in merge sort).

2. Analyse the number of instructions executed in the following iterative algorithm for computing nth Fibonacci numbers as a function of n

```
public static void main(String args[])
{
     int n = Integer.parseInt(args[0]);
     if(n==0)      System.out.println(0);
     else if(n==1)      System.out.println(1);
     else
     {    int fib1 = 0;
          int fib2 =1;
          for(int i=2; i<=n;i=i+1)
          {
              int temp = fib1+fib2;
              fib1 = fib2;
              fib2 = temp;
          }
           System.out.println(fib2);
     }
}
```

   Answer : The algorithm takes cn instructions for some positive constant c.

3. Design an algorithm which computes 3^n using only $c \log n$ instructions for some positive constant c.
   Hint : use the recursive formulation of 3^n carefully.

4. Given an array A which stores 0 and 1, such that each entry containing 0 appears before all those entries containing 1. In other words, it is like $\{0, 0, 0, ..., 0, 0, 1, 1, ..., 111\}$. Design an algorithm to find out the small index i in the array A such that $A[i] = 1$ using $c \log n$ instructions in the worst case for some positive constant c.
   Hint : exploit the idea used in binary search.

5. How many instructions are executed when we multiply $n \times m$ matrix A with $m \times r$ matrix B ?
   Answer : The number of instructions executed is $c\ mnr$ for some positive constant c.
   Hint : Analyse the algorithm for multiplying two matrices as discussed in one of the lecture.