Assignment - 5
yashunhawane@gmail.com

1. Compute and return the square root of x, where x is guaranteed to be a non-negative integer. And since the return type is an integer, the decimal digits are truncated and only the integer part of the result is returned. Also, talk about the time complexity of your Code.

```
let sum = (x)=>{
    let i = 1
    let j = x

    if (x < 2){
        return x
    }

    while(i<j){
        let mid = i +Math.floor((j-i)/2)
        console.log(i,mid,j)

        if(mid*mid == x){
            return mid;

        }
        else if(mid*mid > x){
            j = mid
        }
        else{
            i = mid+1
        }
    } return j-1
}



console.log(sum(3))
```

Time Complexity : O(log(n))
Space Complexity: O(1)

leetcode : 69

2.You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad. Suppose you have n version and you want to find out the first bad one, which causes all the following ones to be bad. Also, talk about the time complexity of your code.

```
return function(n) {
    let i = 1
    let j = n



    while(i<j){
        let mid = i+Math.floor((j -i)/2)


        if(isBadVersion(mid)){
            j = mid
        }
        else{
            i = mid + i
        }
    }return i
};
```

Time Complexity : O(log(n))
Space Complexity: O(1)

leetcode : 278

3.Given a positive integer num, write a program that returns True if num is a perfect square else return False. Do not use built-in functions like sqrt. Also, talk about the time complexity of your code.

```
let square = (num) => {
  let i = 1;
  let j = num;

  if ((num == 1)) {
    return true;
  }

  while (i <=j) {
    let mid = i + Math.floor((j - i) / 2);
    console.log(i,mid,j)
    if (mid * mid == num) {
      return true;
    } else if (mid * mid > num) {
      j = mid - 1;
    } else {
      i = mid + 1;
    }
  }
  return false;
};
```

Time Complexity : O(1)
Space Complexity: O(1)

leetcode : 367