Kelly Williams

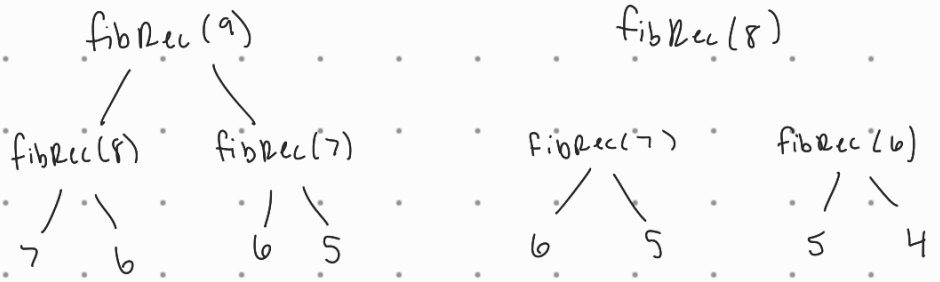
Problem 6

Recursive:

```
int fibRec(int n){
    if(n<=0) return 0;
    if(n==1) return 1;
    return fibRec(n-1)+fibRec(n-2);}
```

$T(n) = T(n-1) + T(n-2) + O(5)$

let $n=10$



At each function call, fibRec gets called 2 times ⇒ exponential increase

$$\boxed{T(n) = O(2^n)}$$

Non-Recursive:

```
int fibLoop(int n){
    if(n<=0) return 0;
    if(n==1) return 1;
    int fib1=1, fib2=0, fi=fib1+fib2;
    for(int i=2; i<n; i++){
        fib2=fib1;
        fib1=fi;
        fi=fib1+fib2;}
    return fi;}
```

non recursive - lots of O(1) statements ⇒ don't rely on input size.

For loop approaches size N so $\boxed{T(n) = O(N)}$