PROBLEMS: UNIT 1

INTRODUCTION TO THE ALGORITHMICS

JAVIER GARCÍA JIMÉNEZ AND ISABEL MARTÍNEZ GÓMEZ

EXERCISE 3

$$T(n) = 1 + \sum_{i=x}^{y} 1 + 1 + \max\left\{1.1 + \sum_{i=x}^{y}\left(\sum_{j=3x}^{3y} 1\right) + 1 + T\left(x + \frac{y-x}{2}\right) + 1\right\}$$

$$T\left(x + \frac{y-x}{2}\right) = T\left(\frac{2x+y-x}{2}\right) = T\left(\frac{x-y}{2}\right) = T(n/2)$$

$$T(n) = 1 + n + 1 + 1 + 3n^2 + 1 + T(n/2) + 1$$

$$T(n) = 3n^2 + n + 5 + T(n/2)$$

$$n = 2^k$$

$$T(2^k) = 3\cdot(2^k)^2 + 2^k + 5 + T(2^k/2)$$

$$T(2^k) = 3\cdot 4^k + 2^k + 5 + T(2^{k-1})$$

$$T(2^k) = x^k$$

$$x^k - x^{k-1} = 3\cdot 4^k + 2^k + 5$$

$$x^{k-1}(x-1) = 3\cdot 4^k + 2^k + 5$$

Homogenous equation

$$x^{k-1}(x-1) = 0$$

Roots: $x = 1$ $\qquad x^{(H)} = A$

Particular equation

$$x^{k-1}(x-1) = 3\cdot 4^k + 2^k + 5$$

Roots: $x = 4,\ x = 2,\ x = 1$ $\qquad x^{(P)} = B\cdot 4^k + C\cdot 2^k + D\cdot k$

$x = A + B\cdot 4^k + C\cdot 2^k + D\cdot k$ $\qquad n = 2^k \rightarrow k = \log_2 n$

$$T(n) = x = A + B\cdot n^2 + C\cdot n + D\cdot \log_2 n \longrightarrow O(n^2)$$

Complexity of $O(n^2)$.

Complexity of $O(n^2)$

EXERCISE 5

```
public boolean isPrime (int n)
{
    boolean b=true;
    for (int i=2;i<n;i++)
    {
        if (n%i==0){
            b=false;
        }
    }
    return b;
}
```

$$T(n) = 1 + \sum_{i=1}^{n} \overbrace{1 + max\{1,0\}}^{if}$$

$$T(n) = 1 + \sum_{i=1}^{n} 1 + 1$$

$$T(n) = 1 + 2n \longrightarrow O(n)$$

Complexity of O(n)

EXERCISE 8

```
public int inverseNumber (int n)
{
    String numberInv="";
    if (n/10==0){
        numberInv+=String.valueOf(n%10);
    }else{
        numberInv+=String.valueOf(n%10)+inverseNumber(n/10);
    }
    return Integer.parseInt(numberInv);
}
```

$T(n) = 1 + 1 + max\{1, 1 + T(n-1)\} = 2 + 1 + T(n-1)$

$T(n) = 3 + T(n-1)$

$T(n) = x^n$

$x^n = 3 + x^{n-1}$

$x^n - x^{n-1} = 3$

$x^{n-1}(x-1) = 3$

HOMOGENOUS EQUATION

$x^{n-1} \cdot (x-1) = 0$

ROOTS: $x = 1$

$x^{(H)} = A \cdot 1^n = A$

PARTICULAR EQUATION

$x^{n-1} \cdot (x-1) = 3$

ROOTS: $x = 1$

$x^{(P)} = (B)n = Bn$

↳ we multiply by n so that the constant A is not the same as B.

$\boxed{x = A + Bn = T(n) \longrightarrow O(n)}$

Complexity of O(n)

EXTRA EXERCISE: EXERCISE 7

```java
public final class Exercise7 {
    Scanner in = new Scanner(System.in);
    int n;
    int nPrimes=0;
    int nPerfects=0;

    public Exercise7()
    {
        askNumberUser();
        exerciseFunction();
        System.out.println("There are "+nPrimes+" prime numbers and "+nPerfects+" perfect numbers");
    }

public void askNumberUser(){
    System.out.println("Introduce a number: ");
    this.n = Integer.valueOf(in.nextLine());

    while(n<0)
    {
        System.out.println("The number must be positive, introduce a number again: ");
        this.n=Integer.valueOf(in.nextLine());
    }
}

public boolean isPrime (int n)
{
     boolean b=true;
     for (int i=2;i<n;i++)
     {
          if (n%i==0) b=false;
     }
     return b;
}

public boolean isPerfect(int n)
{
     int sum=0;
     for(int i=1;i<n;i++)
     {
          if(n%i==0) sum=sum+i;
     }
     return sum==n;
}

public void exerciseFunction()
{
     for(int i=1;i<n;i++)
     {
          if(isPrime(i)) nPrimes++;
          if(isPerfect(i)) nPerfects++;
     }
}
```

$$T(n) = \sum_{i=1}^{n} \left(1 + \underbrace{\left(\sum_{i=1}^{n} 1 + 1 \right)}_{isPrime()} + 1 + 1 + \underbrace{\left(\sum_{i=1}^{n} 1 + 1 \right)}_{isPerfect()} + 1 \right)$$

$$T(n) = \sum_{i=1}^{n} (4 + 2n + 2n)$$

$$T(n) = \sum_{i=1}^{n} (4 + 4n)$$

$$T(n) = 4n + 4n^2 \longrightarrow \underline{O(n^2)}$$

The complexity of the exerciseFunction is of O(n^2)