Design & Analysis of Algorithms

Assignment #01

Name: AOUN - HAIDER

ID: FA21-BSE-153

(Not registered)

Let

$T_1(n)$ for "bar" function

$T_2(n)$ for "foo" function

$T(n)$ for main function

```
bar(a){                                          Number of steps
    for(int i=0; i<n; i++)                       ——— n
        for(int j=0; j<i; ++j)                   ——— Σ_{i=0}^{n} t_i
            a=a*(i+j);
    return a;                                    ——— Σ_{i=0}^{n} t_i
}                                                ——— 1
```

Analysis:

$$T_1(n) = n + \sum_{i=0}^{n} t_i + \sum_{i=0}^{n} t_i + 1$$

$$= n + \frac{n(n+1)}{2} + n + \frac{n(n+1)}{2} + n + 1$$

$$= 3n + 2.\frac{n(n+1)}{2} + 1$$

$$= 3n + n^2 + n + 1$$

$$T_1(n) = n^2 + 4n + 1 \quad \text{or} \quad O(n^2)$$

$T_1(n) = n^2 + 4n + 1$

2)

```
foo(a) {                                   Num of steps
    for( int i=1; i<=n*n; ++i)        ——  n^2
        a = a*i;                      ——  n^2
    return a;                         ——  1
}
```

$T_2(n) = n^2 + n^2 + 1 \Rightarrow 2n^2 + 1$

3) (Main part)

```
                                                  Num of ste
1. for (int i=0; i<n; ++i) {          ——————  n
2.     if( i%2 != 0)                  ——————  n
3.         sum += foo(i) + foo(i+1);  ——  n(T2 + T2)
       else
4.         sum += foo(i) + bar(i);    ——  n(T1 + T2)
   }
```

3) $T_2 + T_2 = 2T_2 = 2(n^2 + 4n + 1) = 2n^2 + 8n + 2$

$n(T_2 + T_2) = 2nT_2 = 2n^3 + 8n^2 + 2n$

4) $n(T_1 + T_2) = n[(2n^2 + 1) + (n^2 + 4n + 1)]$

$= n(3n^2 + 4n + 2)$

$= 3n^3 + 4n^2 + 2n$

$$T(n) = n + n + \text{Max}(2n^3 + 8n^2 + 2n,\ 3n^3 + 4n^2 + 2n)$$

$$= 2n + 2n^3 + 8n^2 + 2n$$

$$\boxed{T(n) = 2n^3 + 8n^2 + 4n}$$

By droping constants & lower order terms in terms of Big-oh

$\Rightarrow O(n^3)$