Homework 1

CS201: Data Structures and Algorithms, Fall 2021
Due: 11:59pm 5 Nov., 2021

Objective:

Calculation of time complexities of algorithms/programs using Big-O notation.

Instructions:

- Submit a clean and clear solution through **Moodle LMS.**
- Only PDF file format will be accepted.
- You may type or hard write (then scan) your solution.
- **Remember**: *If I can't read ... I can't grade.*

Problem 1 (36 Points):

1. (24 pts) Complete the below table, then
2. (12 pts) Sort the last column in ascending order.

Expression	**Dominating Term(s)**	$O(\cdot)$
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	$O(n^3)$
$10^{1000} + 2^{1000} + 4^{200} + 1$		
$5n + n^{1.5} + 3n \log n$		
$\sqrt[3]{n^9} + 10^{10^{10^{10^{10}}}}$		
$n! + 2^n + n \log n$		
$n! + n^n + n^{\log n}$		
$2^{3^n} + 3^{2^n} + 5^n$		
$\sqrt[2]{n} + \log n$		
$0.003 \log n + \log(\log n)$		
$\log_2 n + \log_3 n + \log_5 n$		
$\sum_{i=1}^{10} i * n^i$		
$\prod_{i=1}^{4} n^i$		
$\sum_{i=2}^{10} 30 \log i$		

Problem 2 (32 points):

For each of the below functions, calculate the time complexity using Big-0 notation. SHOW ALL YOUR STEPS.

Function A:

```
void A(){
    cout<<"Hello World"<<endl;
}
```

Function B:

```
void B(int n){
    int i = 0;
    int sum = 0;
    while (i < n){
        sum += i;
        i += 1;
    }
}
```

Function C:

```
void C(int n, int m){
    int i = 0;
    int sum = 0;
    while (i < n){
        int j = 0;
        while (j < m){
            sum += j;
            j += 1;
        }
        i += 1;
    }
}
```

Function D:

```
void D(int n, int m, int l){
    int i = 0;
    int sum = 0;
    while (i < n){
        int j = 0;
        while (j < m){
            int k = 0;
            while( k < l ){
                sum += k;
                k += 1
            }
            j += 1;
        }
        i += 1;
    }
}
```

Function E:

```
int E(){
    int n;
    cout<<"Enter␣a␣positive␣number:";
    cin>>n;
    int i = 0;
    int sum = 0;
    while( i < n ){
        sum += i;
        i += 1;
    }
    return sum;
}
```

Function F:

```
void F(int n){
    int sum = 0;
    for(int i = 0; i < n; i++){
        for(int j = i; j < n; j++){
            sum += 1;
        }
    }
}
```

Function G:

```
void G(int n){
    int sum = 0;
    for(int i = 1; i < n; i = i*2){
        sum += 1;
    }
}
```

Function H:

```
void H(int n){
    int sum = 0;
    for(int i = n; i > 0; i = i/2){
        sum += 1;
    }
}
```

Problem 3 (32 points):

For each of the below functions, calculate the time complexity in Big-0 notation. SHOW ALL YOUR STEPS.

Function A1:

```
void S(int n){
    for(int i = 1; i < n; i = i*2){
        cout<<"i␣=␣"<<i;
    }
}
void L(int n){
    for(int i = 0; i < n; i = i + 2){
        cout<<"L.␣i␣=␣"<<i;
    }
}
void M(int n){
    while(n > 0){
        cout<<"M.␣i␣=␣"<<i;
        n = n/5;
    }
}
void A1(int n){
    S(n);
    L(n);
    M(n);
}
```

Function B1:

```
void B1(int n){
    int sum = 0;
    for(int i = 1; i < n; i = i*2){
        sum += 100;
        if( sum > 500 ){
            break;
        }
    }
}
```

Function C1:

```
int C1(int n){
    if( n == 0)
        return 0;
    return C1(n-1) + n;
}
```

Function D1:

```
int D1(int n){
    if(n == 0)
        return 0;
    return 2*C1(n-1) + n;
}
```

Function E1:

```
int E1(int n){
    if(n == 0)
        return 0;
    return C1(n-1) + C1(n-1) + n;
}
```

Function F1:

```
void F1(int n){
    if(n == 0)
        return 0;
    for(int i = 0; i < n; i++){
        cout<<n<<","<<i;
    }
    F1(n-1);
}
```

Function G1:

```
void G1(int n){
    if(n == 0)
        return 0;
    return G1(n/2) + n;
}
```