วิชา 204451 Algorithm Design and Analysis

รหัสนักศึกษา 6305106600 ชื่อสกุล ศิรประภา เพชรภูสินธนา ตอนที่ 2

Assignment 1 (อ. 28 มิ.ย. 65) : Analyzing the time complexity กำหนดส่งงาน : จ 4 ก.ค. 65 23.59 น.

ให้นักศึกษา

1. เขียนคำตอบตามโจทย์กำหนดด้วยลายมือ แล้วถ่ายรูป (นามสกุล .jpg) หรือไฟล์ pdf ส่งที่เว็บส่งการบ้านภาควิชาฯ
2. ตั้งชื่อไฟล์ในรูปแบบ assign_xx_id เมื่อ x คือหมายเลข Assignment และ id คือ รหัสนักศึกษา (กรณีส่งหลายไฟล์ให้ตั้งชื่อเป็น assign_01_id_a.jpg โดย a หมายถึง ลำดับไฟล์ แล้วทำการ zip รวมทุกไฟล์ส่งในงาน Assignment เดียวกันด้วยชื่อ assign_01_id.zip แทน)
3. ส่งงานภายในวันเวลาที่กำหนด หากส่งเลยกำหนดให้ชี้แจงเหตุผลกับอ. ประจำ section (พิจารณาคะแนนตามเหตุผล)

```
int Function1( const int A[ ], int N ) {
/* 1*/ MaxSum = 0                          c1
/* 2*/ for i = 1 to N do                   c2·n
/* 3*/   sum = 0                           c3·n
/* 4*/   for j = i to N do                 c4·(n-i+1)·n
/* 5*/     sum += A[ j ]                   c5·(n-i+1)·n
/* 6*/     if (sum>MaxSum )                c6·(n-i+1)·n
/* 7*/       MaxSum = sum                  c7·(n-i+1)·n
/* 8*/ return MaxSum                       c8
}
```

แทน n-i+1 ด้วย n เพราะ i=1 คือ worst-case

T(n)= $(c_4+c_5+c_6+c_7)n^2 + (c_2+c_3+c_4+c_5+c_6+c_7)n + (c_1+c_8)$ = O (n^2)

```
int  CALLEDFUNC2(int A [], int n) {
/*1*/  sum = 0, i = 1                     c1
/*2*/  while i <= n  do                   c2·log2 n
/*3*/       A[i-1]= i*i*i                 c3·log2 n
/*4*/       sum=sum+A[i-1]                c4·log2 n
/*5*/       i = i*2                       c5·log2 n
/*6*/   return sum                        c6
} // O(log n) เพราะ (c2+c3+c4+c5) log2 n + (c1+c6)
```

```
void  Function2(int A[], int n) {
       unsigned int i,sum
/*6*/  sum = CALLEDFUNC2(A, n)            c7·O(log n)
/*7*/  for i = 1 to n-1 do                c8·(n-1)
/*8/       print(A[i-1])                 c9·(n-1)
/*9*/  print(A[n-1],sum)                  c10
}
```

T(n)= $(c_7+c_8+c_9)n + c_7 \log n + (c_6-c_8-c_9)$ = O (n)

```
Function3(int, int m, int s[],int A[,]) {
/*1*/ for i = 1 to m do                   c1·m
/*2*/      s[i] = 0                       c2·m
/*3*/ for i =1 to m do                    c3·m
/*4*/      for j =1 to n do               c4·m·n
/*5*/          s[i] = s[i] + A[j,i]       c5·m·n
}
```

T(n)= $(c_4+c_5)mn + (c_1+c_2)n$ = O (mn)

```
int sum Function4( int n ) {
/*1*/ k = n*n*n                           c1
/*2*/ sum=0                               c2
/*3*/ while k > 1 do                      3c3 log3 n
/*4*/   for j = k to sqrt(n) do           c4·3log3 n·j
/*5*/     for i = 1  to n                 c5·3log3 n·j·n
/*6*/       sum = sum+i*j*k               c6·3log3 n·j·n
/*7*/   k= k/3                            c7·3log3 n
/*8*/ return sum                          c8
}
```

แทน j $= n^3 - \sqrt{n} + 1 = O(n^3)$

$3(c_5+c_6)\log_3 n \cdot n^4 + 3(c_4+c_7)\log_3 n \cdot n^3 + 3(c_3+c_7)\log_3 n + (c_1+c_2+c_8)$

T(n)= O $(n^4 \cdot \log n)$

```
void  Function5(int n) {
/*1*/  i=1                                c1
/*2*/  while i < n                        c2·log2 n
/*3*/    j = 256                          c3·log2 n
/*4*/    while j > 1                      c4·log2 n·8
/*5*/      for k = 0 to n step 2          c5·log2 n·8·n/2
/*6*/        print("*")                  c6·log2 n·4n
/*7*/      j /= 2                         c7·log2 n·8
/*8*/    j = 1                            c8·log2 n
/*9*/    while j < n                      c9·log2 n·log2 n
/*10*/     print("*")                     c10·log2 n·log2 n
/*11*/     j *= 2                         c11·(log2 n)^2
/*12*/ i *= 2                             c12·log2 n
}
```

$\log_2 256 = 8$

$(c_5+c_6)\cdot 8 \cdot \frac{n}{2} \log_2 n + (8c_7+c_8+c_4+c_2+c_3+c_{12})\log_2 n + (c_9+c_{10}+c_{11})(\log_2 n)^2 + c_1$

T(n)= O $(n \log n)$

note: $O(n \log n) > O((\log n)^2)$

วิชา 204451 Algorithm Design and Analysis

รหัสนักศึกษา 6305210600 ชื่อสกุล ศิริศักดา เพชรคูณวิชา ตอนที่ 2

```
void Function6(int n) {
/*1*/ i=n                          c1
/*2*/ while i > 0                  c2 · log2 n
/*3*/   j = 1                      c3 · log2 n
/*4*/   while j <= n               c4 · log2 n · log2 n
/*5*/     for k = 0 to n step 2    c5 · log2 n · log2 n · n/2
/*6*/       print("*")            c6 · (log2 n)^2 · n/2
/*7*/     j *= 2                   c7 · (log2 n)^2
/*8*/   i /= 2                     c8 · log2 n
}
```

T(n) = $\frac{(c_5+c_6+)}{2} n (\log_2 n)^2$ $+ (c_4+c_7)(\log_2 n)^2 + (c_2+c_3+c_8)\log_2 n + c_1$ = O $(n(\log n)^2)$

```
void Function7(int n) {
/*1*/ for i = n downto 1 do        c1 · n
/*2*/   j = n                      c2 · n
/*3*/   while j > 0                c3 · n
/*4*/     for k = 0 to n step 2    c4 · n/2 (n+1)
/*5*/       print("*")            c5 · n/2 (n+1)
/*6*/     j *= 2                   c6 · n
}
```

T(n) = $\frac{1}{2}(c_4+c_5) n(n+1) + (c_1+c_2+c_3+c_6)n$ = O (n^2)

```
void Function8(int n) {
/*1*/ for i = n downto 1 do        c1 · n
/*2*/   j = 1                      c2 · n
/*3*/   while j <= n               c3 · n log2 n
/*4*/     for k = 0 to j           c4 (n(log2 n)+Σj)
/*5*/       print("*")            c5 (n log2 n + n − 1)
/*6*/     j *= 2                   c6 · n log2 n
}
```

T(n) = $(c_4+c_5)(n\log_2 n) + (c_4+c_5+c_7)n + (-c_4-c_5)$ = O $(n \log n)$

$j = 1, 2, 4, \ldots, n \rightarrow n = 2^k \rightarrow k = \log_2 n$

$\Sigma j = \frac{2^{\log_2 n} - 1}{2-1} = n-1$

```
void Function9(int n) {
/*1*/ sum = 0                      c1
/*2*/ for(i=0;i<sqrt(n)/2;i++)     c2 · √n/2
/*3*/   sum++                      c3 · √n/2
/*4*/ for j=0;j<sqrt(n)/4;j++)     c4 · √n/4
/*5*/   sum++                      c5 · √n/4
/*6*/ for k=0;k<sqrt(n)/8;k++)     c6 · √n/8
/*7*/   sum++                      c7 · √n/8
}
```

T(n) = $\left(\frac{c_2+c_3}{2} + \frac{c_4+c_5}{4} + \frac{c_6+c_7}{8}\right)\sqrt{n}$ + c_1 = O $(\sqrt{n})$

```
void Function10(int n) {
/*1*/ sum = 0                      c1
/*2*/ for(i=0;i<sqrt(n)/2;i++)     c2 · √n/2
/*3*/   for(j=i;8+i;j++)           c3 · (√n/2)(8)
/*4*/     for(k=j;k<8+j;k++)       c4 · (√n/2)(8^2)
/*5*/       sum++                  c5 · (√n/2)(8^2)
}
```

T(n) = $\left(\frac{c_2}{2} + 4c_3 + 32c_4 + 32c_5\right)\sqrt{n}$ + c_1 = O $(\sqrt{n})$

```
void Function11(int n) {
/*1*/ sum = 0                      c1
/*2*/ for (i = 0; i < n; i++)      c2 · n
/*3*/   for (j = 0; A[j] <> i; j++)  c3 · (n + n(n−1)/2)
/*4*/     sum++                    c4 · (n + n(n−1)/2)
}
```

not equal to

T(n) = $\left(\frac{c_3}{2} + \frac{c_4}{2}\right)n^2 + (c_2+c_3+c_4)n$ $+ (c_1 - c_3 - c_4)$ = O (n^2)

```
long gcd(long m, long n){
/*1*/ while(n! = 0)                 c1 · (log2 m + log2 n)
/*2*/     long rem = m%n           c2 · (log2 mn)
/*3*/     m = n                    c3 · (log2 mn)
/*4*/   n = rem                    c4 · (log2 mn)
}
```

indentation error

T(n) = $(c_1+c_2+c_3+c_4)\log_2 mn$ = O $(\log mn)$