UTM BACHELOR OF COMPUTING SCIENCE (DATA ENGINEERING) COMPUTING PROGRAMME

DATA STRUCTURE AND ALGORITHM

(SECJ2013-02)

ASSIGNMENT 4
ALGORITHMS EFFICIENCY

NAME = 1) NUR HADIRAH MUNAWARAH BINTI ROZMIZAN (A19EC0201)

2) NUR ALEEYA SYAKILA BINTI MUHAMAD SUBIAN (A19EC0127)

LECTURER = DR RUHAIDAH BINTI SAMSUDIN

SUBMISSION DATE = 12/11/2020

SIGNATURE AND SCORE =

Question 2

a)

Statements	Number of steps
For (i=2 ; i<n;i++)	n-2+1+1 =n-2
Cout << I;	1.(n-2) =n-2
For (j=1;j<=n;j=j*2)	1+$\log_2 n$
Cout << j;	1.(1+$\log_2 n$) =1+$\log_2 n$
Total execution	2(n-2)+2($1 + \log_2 n$)
Time complexity	**O($\boldsymbol{\log_2 n}$)**

b)

Statements	Number of steps
Int i=1	1
While (i<=n)	n-1+1 =n
{ for (j=1;j<=I;j++)	n =n
Cout << I ;	n.(n) =n^n
Cout << j;	1.(n^n) =n^n
I++ ; }	1.(n) =n
Total execution	1+2n+n^n
Time complexity	**O(n^n)**

c)

Statements	Number of steps
Int I	1
J=1	1
For (i=1;i<=n;i++) {	n-1+1 =n
J=j*2; }	n.$\log_2 n$ =n$\log_2 n$
For (i=1;i<=j;i++) {	2^n
Cout << j <<"\n";	1.(2^n) =2^n
Total execution	2+n+ n$\log_2 n$+2(2^n)
Time complexity	**O(2^n)**

d)

Statements	Number of steps
For (a=3;i<=n;a++)	n-3+1 =n-2
For (b=1;b<n;b++)	(n-2).n =n^2-2n
Cout << a;	1.(n2-2n) =n^2-2n
Total execution	n-2+2(n^2-2n)
Time complexity	**O(n^2)**

Question 3

Code Segment

```
for(int i = 1; i <= 5; ++i)
{
        for(int j = 1; j <= i; ++j)
            {
                cout << "# ";
             }
      cout << "\n";
}
```

Statements	Number of steps
for(int i = 1; i <= 5; ++i){	n-1+1 =n
for(int j = 1; j <= i; ++j)	n.n $=n^2$
cout << "# "; }	1. n^2 $=n^2$
cout << "\n";	1.n =n
Total execution	2n + 2(n^2)
Time complexity	**O(n^2)**

Question 4

STATEMENT	Number of Steps
Int i=0	1
For (i=4 ;i<=n;i=i*4)	1+$\log_4 n$
Cout << "Current I value is " << I << endl;	$1.(1+\log_4 n)$ =$(1+\log_4 n)$
Cout << "Current n value is " << I << endl;	=1
Total steps	$2+2(1+\log_4 n)$
Algorithm complexity	**O($\log_2 n$)**

Calculation

$\log_4 n$ = $\frac{\log_2 n}{\log_2 4}$

=$\log_2 n$ /2

=**O($\log_2 n$)**

Question 5

STATEMENT	Number of Steps
For (i=4;i<n;i++)	n-4-1+1 =n-4
Cout << I;	1.(n-4) = n-4
For (j=1;j<=n;j=j*2) {	1+($\log_2 n$)
Cout << j;	1.(1+($\log_2 n$)) =1+($\log_2 n$)
For (k=1;k<=n;k++)	(n-1+1). (1+($\log_2 n$)) =n.(1+$\log_2 n$)
Cout << k }	1. ((n-1+1). (1+($\log_2 n$))) 1.n.(1+$\log_2 n$) = n.(1+$\log_2 n$)
Total steps	2(n-4)+2($\log_2 n$)+2($n \log_2 n$)
Algorithm complexity	**O($\mathbf{nlog_2} \boldsymbol{n}$)**

Question 6

a)

Value of I	Number of streps for in the internal loop For (j=1;j<=I;j++)
1	1= 1 time
2	1,2 = times
3	1,2,3= 3 times
:	:
n-1	1,2,3,...n-2,n-1=n-1 times
n	1,2,3,....n-1,n=n+1 times

B) Total steps = 1+n+(n(n+1)/2)+(n+1)/2)+n

=1+n+$\frac{n^2+n+n^2+n}{2}$+n

=1+n+n^2+n+n

=1+3n+n^2

c) Big O notation = O (n^2)

d) Bubble sort has big O notation of O(n^2)

Question 7

Statement	Number of steps
for (j=1;j<=l;j=j*2)	$\log_2 n$
Cout << “EXCELLENT!”;	1.$(\log_2 n)$
for (i=4;i<=n;i++)	$n-4+1$ =n-3
Cout << “GREAT!!!”;	1.(n-3) =n-3
for (k=1;k<=n;k++)	(n-1+1) =n
Cout <<”**********”;	n.1 =n
Total steps	$2\log_2 n + n - 3 + n - 3 + 2n$ =$2\log_2 n + 4n - 6$
Algorithm complexity	**O($\boldsymbol{\log_2 n}$)**

Question 8

i)

Value of I	**Number of streps for in the internal loop** **For (j=1;j<=I;j++)**
N	n time
n-1	n-1 time
:	:
3	3 times
2	2 times
1	1 time

Ii) Total steps

=n + 1 + (n(n-1)/2)+(n(n-1)/2)+n

=1+2n + $\frac{n^2-n+n^2-n}{2}$

=1+2n + $\frac{2n^2-2n}{2}$

=1+2n+n^2-n

=1+n+n^2

iii) Big O notation = O(n^2)