DSA I – Solving Session

Time Complexity Analysis

by Mahfuz Hasan Reza

b) Discuss the time complexity of the following algorithm. [3]

```
sum=0;
for(i=1; i<=n; i++){
    for(j=1; j<=n; j++){
        sum=sum+i+j;
    }
}
printf("%d", sum);
```

DSA I - Solving Session

Time Complexity Analysis

Previous Question Solve

by Mahfuz Hasan Reza

* Mid Exam, Year: 2023, Trimester: Spring

1 (b)

```
sum=0;                       ——————> 1
for(i=1; i<=n; i++){         ——————> (n+1)
    for(j=1; j<=n; j++){     ——————> n x (n+1)
        sum=sum+i+j;         ——————> n x (n)
    }
}
printf("%d", sum);           ——————> 1
```

Time function, $T(n) = 1 + n + 1 + n^2 + n + n^2 + 1$

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

$\therefore$ Time complexity $= O(n^2)$

b) Discuss the time complexity of the following algorithm. [3]

```
sum=0;
for(i=1; i<=n; i++){
   for(j=1; j<=i; j++){
       sum=sum+i+j;
   }
}
printf("%d", sum);
```

* Mid Exam, Year : 2022, Trimester: Fall

(1)(b)

```
sum=0;                          ——> 1
for(i=1; i<=n; i++){            ——> (n+1)
   for(j=1; j<=i; j++){         ——> (n+1)(n+2)/2 - 1
       sum=sum+i+j;             ——> n(n+1)/2
   }
}
printf("%d", sum);              ——> 1
```

For inner loop:

i :	1	2	3	4	5	....	n	(n+1)
head:	2	3	4	5	6	....	(n+1)	
body:	1	2	3	4	5	....	n	

$\therefore$ head : $1+2+3+4+5+6+\cdots+(n+1)-1$

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

$\therefore$ body : $1+2+3+4+5+\cdots+n$

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

$\therefore$ Time function, $T(n) = 1 + (n+1) + \frac{(n+1)(n+2)}{2} - 1 + \frac{n(n+1)}{2} + 1$

$\therefore$ Time Complexity $= O(n^2)$

b) Discuss the time complexity of the following algorithm. [2]

```
sum=0;
for(i=1; i<=n; i++){
   for(j=1; j<=i; j++){
      sum=sum+i+j;
   }
}
printf("%d", sum);
```

Mid Exam, Year: 2022, Trimester: Spring

(1) (b)

Same as Fall-2022 (Mid Exam) - 1(b)

b) Discuss the time complexity of the following algorithm. [2]

```
sum=0;
for(i=0; i<n; i++){
   scanf("%d", &A[i]);
   j=n-1;
   while(j>=0){
       sum=sum+A[i]+A[j];
       j--;
   }
}
```

Mid Exam, Year : 2021, Trimester : Summer

1 (b)

```
sum=0;  ————————→ 1
for(i=0; i<n; i++){ ——→ (n+1)
   scanf("%d", &A[i]); ——→ n
   j=n-1; ————————→ n
   while(j>=0){ ————→ n × (n+1)
       sum=sum+A[i]+A[j]; → n × n
       j--; ————————→ n × n
   }
}
```

$\therefore$ Time function, $T(n) = 1+(n+1)+n+n+n(n+1)+n^2+n^2$

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

$\therefore$ Time Complexity, $O(n^2)$

4. How many times each of statements and conditions will be executed?

```
k=0;
for( i=0; i<=3; i++){
    sum=0;
    for (j=1; j<4; j++){
        sum=sum+i+j;
        k=k+1;
    }
    k=1;
}
```

Random Problem : (solve)

⊛ How many times?

k=0; → 1 times

for(i=0; i<=3; i++){ → 5 times

sum=0; → 4 times

for(j=1; j<4; j++){ → (4x4) times

sum=sum+i+j; → (4x3) times

k=k+1; → (4x3) times

}

k=1; → 4 times

}

Total Times : 1+5+4+16+12+12+4

= 54 times

```
sum=0;
k=10;
for(i=1; i<=n; ++i){
   for(j=3; j<=n; ++j){
       sum=sum+j;
       k=k-1;
    }
    printf("%d", sum);
}
printf(("%d %d", sum, k);
```

Random Problem: (Solve)

Time complexity

$$\begin{array}{lcl}
\texttt{sum=0;} & \longrightarrow & 1 \\
\texttt{k=10;} & \longrightarrow & 1 \\
\texttt{for(i=1; i<=n; ++i)\{} & \longrightarrow & (n+1) \\
\quad \texttt{for(j=3; j<=n; ++j)\{} & \longrightarrow & n \times (n-1) \\
\qquad \texttt{sum=sum+j;} & \longrightarrow & n \times (n-2) \\
\qquad \texttt{k=k-1;} & \longrightarrow & n \times (n-2) \\
\quad \texttt{\}} & & \\
\quad \texttt{printf("\%d", sum);} & \longrightarrow & n \\
\texttt{\}} & & \\
\texttt{printf("\%d \%d", sum, k);} & \longrightarrow & 1
\end{array}$$

Time function, $T(n) = 1+1+(n+1)+(n^2-n)+(n^2-2n)+(n^2-2n)+n+1$

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

$\therefore$ Time Complexity, $O(n^2)$

```
sum=0;
k=0;
while(k<=n){
   sum=sum+k;
   printf("%d  %d", sum, k);
   k=k+1;
}
printf("%d", sum);
```

Random Problem: (Solve)

Time Complexity

```
sum=0;                          ──────→ 1
k=0;                            ──────→ 1
while(k<=n){                    ──────→ (n+2)
   sum=sum+k;                   ──────→ (n+1)
   printf("%d %d",sum,k);       ──────→ (n+1)
   k=k+1;                       ──────→ (n+1)
}
printf("%d", sum);              ──────→ 1
```

Time function, $T(n) = 1+1+(n+2)+(n+1)+(n+1)+(n+1)+1$

$= 4n+8$

$\therefore$ Time Complexity, $O(n)$