ABES
Engineering College
College Code-032

Data Structure Training
Complexity

Department of Computer Science & Engineering
ABES Engineering College, Ghaziabad

Technical Training (Complexity)

Algorithm: An algorithm is a set of instructions needed in order to execute a task.

Need of Complexity?

1. Run in less time(Time Complexity).
2. Consume less Memory(Space Complexity).

Running Time of a Complexity:

- ✓ Depends on size of input.
- ✓ Running time is measured in terms of number of steps/primitive operations performed.
- ✓ Independent from Machine, OS.

Technical Training (Complexity)

Ex- Addition of all numbers in Array:

```
// Input: int A[N], array of N integers
// Output: Sum of all numbers in array A

int Sum(int A[], int N)
{
   int s=0;
   for (int i=0; i< N; i++)
      s = s + A[i];
   return s;
}
```

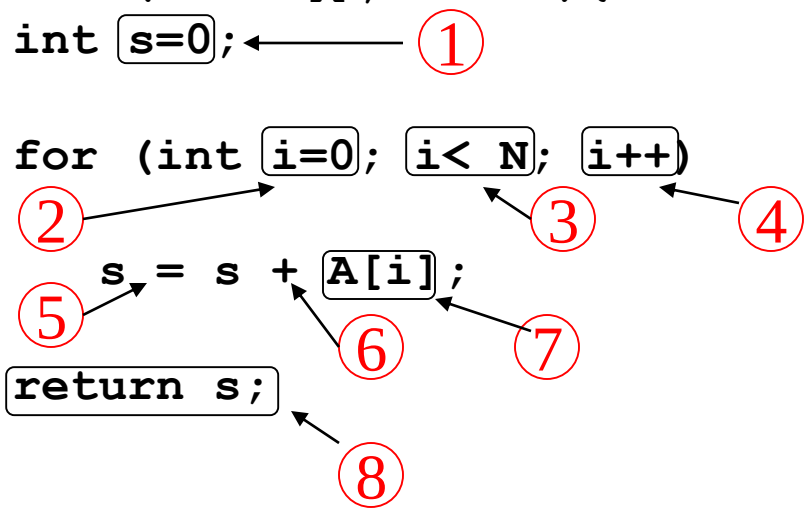
How should we analyse this??

Technical Training (Complexity)

Ex- Addition of all numbers in Array:

```
// Input: int A[N], array of N integers
// Output: Sum of all numbers in array A

int Sum(int A[], int N){
    int s=0;
    for (int i=0; i< N; i++)
        s = s + A[i];
    return s;
}
```



1,2,8: Once

3,4,5,6,7: Once per each iteration of for loop, N iteration

Total: 5N + 3

The *complexity function* of the algorithm is : $f(N) = 5N + 3$

Technical Training (Complexity)

Ex- Addition of all numbers in Array:

// Input: int A[N], array of N integers
// Output: Sum of all numbers in array A

```
int Sum(int A[], int N)
{
  int s=0;
  for (int i=0; i< N; i++)
    s = s + A[i];
  return s;
}
```

How should we analyse this??

1,2,8: Once
3,4,5,6,7: Once per each iteration of for loop, N iteration
Total: 5N + 3
The *complexity function* of the algorithm is : $f(N) = 5N + 3$

Estimated running time for different values of N:

N = 10	=> 53 steps
N = 100	=> 503 steps
N = 1,000	=> 5003 steps
N = 1,000,000	=> 5,000,003 steps

As N grows, the number of steps grow in *linear* proportion to N for this function *"Sum"*

Technical Training (Complexity)

Asymptotic Functions:

1) Big Oh Notations: Upper Bound

2) Omega Notations: Lower Bound

3) Theta Notations: Tighter Bound

Technical Training (Complexity)

Which Notation do we use?

✓To express the efficiency of our algorithms which of the three notations should we use?

✓As computer scientist we generally like to express our algorithms as big O since we would like to know the upper bounds of our algorithms.

✓Why?

✓If we know the worse case then we can aim to improve it and/or avoid it.

Technical Training (Complexity)

Performance Classification

f(*n*)	Classification
1	*Constant*: run time is fixed, and does not depend upon n. Most instructions are executed once, or only a few times, regardless of the amount of information being processed
log n	*Logarithmic*: when *n* increases, so does run time, but much slower. Common in programs which solve large problems by transforming them into smaller problems. Exp : binary Search
n	*Linear*: run time varies directly with *n*. Typically, a small amount of processing is done on each element. Exp: Linear Search
n log n	When *n* doubles, run time slightly more than doubles. Common in programs which break a problem down into smaller sub-problems, solves them independently, then combines solutions. Exp: Merge
n^2	*Quadratic*: when *n* doubles, runtime increases fourfold. Practical only for small problems; typically the program processes all pairs of input (e.g. in a double nested loop). Exp: Insertion Search
n^3	*Cubic*: when n doubles, runtime increases eightfold. Exp: Matrix
2^n	*Exponential*: when n doubles, run time squares. This is often the result of a natural, "brute force" solution. Exp: Brute Force. Note: logn, n, nlogn, n^2>> less Input>>Polynomial n^3, 2^n>>high input

Technical Training (Complexity)

Arrange the following functions in increasing order of their growths? Choose the correct option.
O(n), O(log(n)), O(n(log(n))), O(n^2)?

A. O(n), O(log(n)), O(nlog(n)), $O(n^2)$

B. O(log(n)), O(nlog(n)), $O(n^2)$, O(n)

C. O(n),$O(n^2)$,O(nlog(n)), O(log(n))

D. O(log(n)), O(n), O(nlog(n), $O(n^2)$

Technical Training (Complexity)

Arrange the following functions in increasing order of their growths? Choose the correct option.
O(n), O(log(n)), O(n(log(n))), O(n^2)?

A. O(n), O(log(n)), O(n(log(n))), $O(n^2)$

B. O(log(n)), O(n(log(n)), $O(n^2)$, O(n)

C. O(n),$O(n^2)$,O(n(log(n))), O(log(n))

D. O(log(n)), O(n), O(n(log(n)), $O(n^2)$

Technical Training (Complexity)

Standard Analysis Techniques

- ✓Constant time statements
- ✓Analyzing Loops
- ✓Analyzing Nested Loops
- ✓Analyzing Sequence of Statements

Technical Training (Complexity)

Constant time statements

- ✓ Simple Statement like Assignment, arithmetic expression evaluation
- ✓ requires O(1) time statements.

Example :-

Technical Training (Complexity)

Analyzing Loops

- ✓ **How to Analyze?**
- ✓ How many iteration are performed?
- ✓ How many steps are performed?

Example :-

```
int sum = 0,  j;
for (j=0; j < N; j++)
    sum = sum + j;
```

Technical Training (Complexity)

Analyzing Nested Loops

- ✓ **How to Analyze?(Treat in same manner as done previously)**
- ✓ How many iteration are performed?
- ✓ How many steps are performed?

Example :-

```
int j,k;
for (j=0; j<N; j++)
  for (k=N; k>0; k--)
    sum += k+j;
```

Technical Training (Complexity)

Analyzing Sequence of Loops

- ✓ **How to Analyze?(Treat in same manner as done previously)**
- ✓ How many iteration are performed?
- ✓ How many steps are performed?

Example :-

```
for (j=0; j < N; j++)
    for (k =0; k < j; k++)
        sum = sum + j*k;
for (l=0; l < N; l++)
    sum = sum -l;
```

Technical Training (Complexity)

Analyzing Sequence of Loops

- ✓ **How to Analyze?(Treat in same manner as done previously)**
- ✓ How many iteration are performed?
- ✓ How many steps are performed?

Example :-

```
int sum = 0,  j;
        for (j=1; j < N;  j*2)
            sum = sum + 0;
```

Technical Training (Complexity)

TCS

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = n; i > 0; i--)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O(1)

Technical Training (Complexity)

TCS

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = n; i > 0; i--)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code?
Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O(1)

Technical Training (Complexity)

Capegemini

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = n; i > 0; i /= 2)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O(1)

Technical Training (Complexity)

Capegemini

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = n; i > 0; i /= 2)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O(1)

Technical Training (Complexity)

TCS

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = 0; i < n*2 ; i++)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. $O(n^2)$

Technical Training (Complexity)

TCS

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = 0; i < n*2 ; i++)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code?
Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. $O(n^2)$

Technical Training (Complexity)

Infosys

```
1    #include <stdio.h>
2    int main(void) {
3       int n;
4       for(int i = 2 ; i < n; i++)
5       {
6           for(int j = i; (j > i * 2); j++)
7           {
8               printf("hii");
9           }
10      }
11   }
```

What would be the complexity of the following code?
Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. $O(n^2)$

Technical Training (Complexity)

Infosys

```
1    #include <stdio.h>
2    int main(void) {
3       int n;
4       for(int i = 2 ; i < n; i++)
5       {
6           for(int j = i; (j > i * 2); j++)
7           {
8               printf("hii");
9           }
10      }
11   }
```

What would be the complexity of the following code?
Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. $O(n^2)$

Technical Training (Complexity)

Capegemini

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = 0; i * i < n ; i++)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O (√n)

Technical Training (Complexity)

Capegemini

```
1    int foo(int n) {
2        int c = 0;
3        for (int i = 0; i * i < n ; i++)
4            c += 1;
5        return c;
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O (√n)

Technical Training (Complexity)

Infosys

```
1    int foo(int n) {
2        int i = 1, s = 1;
3        while(s <= n){
4            i++;
5            s = s + i;}
6            return s;
7    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O (√n)

Technical Training (Complexity)

Infosys

```
1    int foo(int n) {
2        int i = 1, s = 1;
3        while(s <= n){
4            i++;
5            s = s + i;}
6            return s;
7    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. O(log(log(n)))

D. O (√n)

Technical Training (Complexity)

Infosys

```
1  int foo ( )
2  {
3      int n;
4      for(int i = 1; i < n; i++)
5          for(int j = 1; j < i; j++)
6              for(int k = 1; k < 100; k++)
7                  printf("HII");
8  }
```

What would be the complexity of the following code? Choose the correct option.

A. $O(n^2)$

B. $O(\log(n))$

C. $O(n(\log(n)))$

D. $O(n^2\log(n))$

Technical Training (Complexity)

Infosys

```
1  int foo ( )
2  {
3      int n;
4      for(int i = 1; i < n; i++)
5          for(int j = 1; j < i; j++)
6              for(int k = 1; k < 100; k++)
7                  printf("HII");
8  }
```

What would be the complexity of the following code? Choose the correct option.

A. $O(n^2)$

B. O(log(n))

C. O(n(log(n)))

D. O (n^2log(n))

Technical Training (Complexity)

TCS

```
1   int foo ( )
2   {
3       int n;
4       for(int i = 1; i < n; i++) {
5           for(int j = 1; j < i*2; j++) {
6               for(int k = 1; k < n/2; k++) {
7                   printf("HII");
8               }
9           }
10      }
11  }
```

What would be the complexity of the following code? Choose the correct option.

A. $O(n^3)$

B. $O(n^4)$

C. O(n(log(n)))

D. $O (n^3 log(n))$

Technical Training (Complexity)

TCS

```
1   int foo ( )
2   {
3       int n;
4       for(int i = 1; i < n; i++) {
5           for(int j = 1; j < i*2; j++) {
6               for(int k = 1; k < n/2; k++) {
7                   printf("HII");
8               }
9           }
10      }
11  }
```

What would be the complexity of the following code? Choose the correct option.

A. $O(n^3)$

B. $O(n^4)$

C. O(n(log(n)))

D. O (n^3log(n))

Technical Training (Complexity)

TCS

```
1    int foo ( )
2    {
3        int n;
4        for(int i = 1 ; i < n ; i = i*2)
5            printf("HII");
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. $O(n^2)$

D. $O(\sqrt{n})$

Technical Training (Complexity)

TCS

```
1    int foo ( )
2    {
3        int n;
4        for(int i = 1 ; i < n ; i = i*2)
5            printf("HII");
6    }
```

What would be the complexity of the following code? Choose the correct option.

A. O(n)

B. O(log(n))

C. $O(n^2)$

D. $O(\sqrt{n})$

Technical Training (Complexity)

Capegemini

```
1   int foo ( )
2   {
3       int n;
4       for(int i = n/2 ; i <= n ; i++) {
5           for(int j = 1 ; j <= n ; j = j*2) {
6               for(int k = 1 ; k <= n ; k = K*2) {
7                   printf("HII");
8               }
9           }
10      }
11  }
```

What would be the complexity of the following code?
Choose the correct option.

A. $O(n^3)$

B. $O(n^4)$

C. $O(n(\log(n)^2))$

D. $O(n^3\log(n))$

Technical Training (Complexity)

Capegemini

```
1   int foo ( )
2   {
3       int n;
4       for(int i = n/2 ; i <= n ; i++) {
5           for(int j = 1 ; j <= n ; j = j*2) {
6               for(int k = 1 ; k <= n ; k = K*2) {
7                   printf("HII");
8               }
9           }
10      }
11  }
```

What would be the complexity of the following code?
Choose the correct option.

A. $O(n^3)$

B. $O(n^4)$

C. $O(n(\log(n)^2))$

D. $O(n^3\log(n))$