Data Structures

SYBTech(CSE)

Unit – 1

Introduction

PPT-5

Prof. Ms. Manisha A. Bhusa
Dept. Of CSE
COE Ambajogai

Unit 1 [6 hrs]

Introduction:

Data,

Data types,

Data structure,

Abstract Data Type (ADT),

Representation of Information,

Characteristics of algorithm,

Program,

Analyzing programs.

Objective

Be familiar with analysis of Algorithm

(Time Complexity).

Time Complexity:

- T(P) = Compile Time(P) + Execution Time(P)

  Where P: Program

- Run Time is denoted by tp : Instance characteristics
- Tp(n) can be obtained experimentally:

  1. Type a program
  2. Compile
  3. Execute

- **Problem in experimental approach:**

In a multiuser system, the execution time depends on:

1. **System load, no. of other programs running on computer at the time program P is running.**
2. **The characteristics of these other programs.**

   so on.

Obtain total number of program steps:

Two ways.

1. Introduce a new variable(global), count, into the program.
2. Tabular method.

```
Count:=0;
Algorithm Swap(a, b)
{
  a := a+b;
  count ++;
  b := a-b;
  count ++;
  a := a-b;
  count ++;
  Display a, b;
  count ++;
}
Total steps := 4
```

1. Introduce a new variable(global), count:

```
Algorithm Sum (a, n)
{
        s := 0.0;
        count:=count + 1;
        for i:= 1 to n step 1 do
        {
                count:=count + 1;
                s:= s + a[i];
                count:=count + 1;
        }
        count:=count + 1;
        count:=count + 1;
        return s;
}
```

```
Algorithm Sum (a, n)
{
   s := 0.0;
   for i:= 1 to n step 1 do
          s := s + a[i];
  return s
}
```

```
Algorithm Sum (a, n)
{
  s := 0.0;
  count:=count + 1;
  for i:= 1 to n step 1 do
  {
     count:=count + 1;
     s:= s + a[i];
     count:=count + 1;
  }
  count:=count + 1;
  count:=count + 1;
  return s;
}
```

Simplified Algorithm:

```
Algo Sum(a, n)
{
   for i:= 1 to n step 1 do
         count := count +2;
   count := count +3;
}
```

count= 2n + 3

Therefore,

Total steps: 2n + 3

Total steps: 2n + 3

If n=10

Then Total steps = 23

If 1 step require 1 Nanosecond for execution

then total time required = 23 Nanoseconds

```
Algorithm RSum (a, n)
{
    count:=count +1
    if (n<=0) then
    {
        count:=count +1
        return 0.0;
    }
    else
    {
        count:=count +1
        return a[n] + RSum(a, n-1);
    }
}
```

Recurrence Relation:

$$TRSum(n) = \begin{cases} 2 & ; \text{ if } n=0 \\ 2 + tRSum(n-1) & ; \text{ if } n>0 \end{cases}$$

Recurrence Relation:

$$
TRSum(n) \begin{cases} = 2 & \text{if } n=0 \\ = 2 + tRSum(n-1) & \text{if } n>0 \end{cases}
$$

Solve recurrence relation by repeated substitution:

$$
\begin{aligned}
tRSum(n) &= 2 + tRSum(n-1) \\
&= 2 + 2 + tRSum(n-2) \\
&= 2(2) + tRSum(n-2) \\
&= 3(2) + tRSum(n-3) \\
&\quad --- \\
&= n(2) + tRSum(n-n) \\
&= 2n + 2
\end{aligned}
$$

Step Count = 2n + 2

This is called Linear Time Algorithm as run time grows linearly in n.