Data Structures

SYBTech(CSE)

Unit – 1

Introduction

PPT-4

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 basics of Algorithm and its criteria.

Algorithm:

This word comes from name of the Persian author, who wrote a book on Mathematics.

~~1. Start~~	1. Start
~~2. Display c~~	2. Read a, b
~~3. c = a+b~~	3. c = a+b
~~4. Read a, b~~	4. Display c
~~5. Stop~~	5. Stop

Definition:

An Algorithm is a set of instructions if accomplishes, perform a specified task.

All algorithms should satisfy following criterion:

1. **Input:**

   Zero or more

2. **Output:**

   At least one

3. Definiteness:

Instruction should be clear & unambiguous.

Ex: Add 6 or 7 to x

Compute 5/0

4. Finiteness:

Terminate after a finite number of steps.

Ex: Chess

5. Effectiveness:

Instruction must be very basic enough so that it can be solved by a person using only pencil and paper.

Data: integer

Program Analysis:

1. Space Complexity
2. Time Complexity

1. **Space complexity:** Amount of memory algorithm needs to run for completion.
2. **Time complexity:** Amount of computer time algorithm needs to run for completion.

Space Complexity:

- Space = Fixed part + Variable part of an algorithm.

1. **Fixed part:** Independent of characteristics of the I/Ps and O/Ps.

   It includes space for

   a. Code.

   b. Simple variables and fixed size component variables.

   c. Constants.

2.Variable part consists of space needed by component variables whose size is dependent on:

a. Particular problem instance being solved.

b. Space needed by referenced variables.

c. Recursion stack space.

Space requirement for algorithm P is:

$$S(P) = C + S_p$$

where, C: Constant

S_p: Instance characteristics

Algorithm abc (a, b, c)

{

```
return a + b + b * c + (a + b - c) / ( a + b) + 4.0;
```

}

- **Characterized by 1 constant, 4.0 &**
  values of a, b & c
- **Assume each value require 1 word space.**
- **No instance characteristics.**

S(abc) = C + Sabc

Therefore **Sabc = 0**

C = 4

S(abc) >= 4 + 0

Sum of n numbers. (Iterative Method)

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

- Characterized by **n and a.**
  - n: number of elements
  - a: array of floating point numbers.
- **Variables:** n, i, s (3 words)
- **n words** for a[ ]

  S(Sum) >= n + 3

Sum of n numbers. (Recursive)

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

n = 3 a = {3, 5, 4}

Rsum(a,3)

{

  if(3<=0)

   False

  else

   return $\overbrace{\overset{4}{a[3]}+\overset{8}{Rsum(a,2)}}^{12}$

}

Rsum(a,2)

{

  if(2<=0)

   False

  else

  return $\overbrace{\overset{5}{a[2]}+\overset{3}{Rsum(a,1)}}^{8}$

}

Rsum(a,1)

{

  if(1<=0)

   False

  else

   return $\overbrace{\overset{3}{a[1]}+\overset{0}{Rsum(a,0)}}^{3}$

}

Rsum(a,0)

{

  if(0<=0)

   return 0

}

- **The recursion stack space includes:**

  Space for **1. formal parameters.**

  2. local variables.

  3. return address.

- **Assume return addr. requires 1 word of memory.**
- **RSum requires 3 words: 1. Space for the value of n**

  2. Return address

  3. A pointer to a[ ]

- **The depth of the recursion is n+1.**

  Therefore, **S(RSum) >= 3(n+1)**