Lab Manual

Module- 03
Course Title: Sessional based on CSE 1201
Course No. : CSE 1202

Experiment No. 3

Name of the Experiment: Arrays, Records and Pointers

Duration: 1 cycle

Background Study: Chapter 4 (Theory and Problems of Data Structures Written by Seymour Lipschutz)

Problem I: Find the Complexity of the bubble Sort algorithm.
Algorithm3.1: (Given a nonempty array A with n numerical values. This algorithm sorts the values)

      1. Set n := length(A)
      2. swapped = false
      3. Repeat i = 1 to n-1 by 1
      4.   If A[i-1]>A[i] then:
                    Swap (A[i-1], A[i]) and swapped = true.
              [End of If Structure]
            [End of Repeat 4 loop]
      5. If swapped = true then:
                    Go to step 2
            [End of If structure]
      6. Exit

Complexity: The complexity of the above algorithm is $O(n^2)$. (Let c= 2)
Complexity Table:

n	f(n) [from Program, Count Statement]	cg(n) [Theoretical]
10		
20		

Graph: Draw a Graph.

Problem II: Linear Search.
Algorithm3.2: (Given a nonempty array A with n numerical values and a specific x of information is given. This algorithm finds the location LOC of x in the array A or Sets LOC=-1)

      1. Set K:=1, LOC:=-1
      2. Repeat steps 3 and 4 while LOC = -1 and K≤n
      3.   IF x = A[K] then:
                    Set LOC:=K.
          [End of If structure]
      4.   K:=K+1.
          [End of step 2 loop]
      5. If LOC = -1 then: Write: x is not in the array A.
         Else: Write: LOC is the location of x
         [End of If structure]
      6. Exit

Complexity: The worst case complexity is C(n) =n and the average case complexity is C(n) = (n+1)/2.

Problem III: Binary Search.
Algorithm3.3: (Given a sorted array A with n numerical values and a specific x of information is given. This algorithm finds the location LOC of x in the array A or Sets LOC=-1)

      1. Set LB:=1, UB:=n, MID = INT((LB+UP)/2) and LOC:=-1
      2. Repeat steps 3 and 4 while LOC = -1 and LB≤UB
      3.   IF x < A[MID] then:
                    Set UB:=MID-1.
         Else If x > A[MID] then:

Set LB:=MID+1.
Else:
Set LOC:=MID
[End of If structure]
4.   MID = INT((LB+UP)/2).
[End of step 2 loop]
5.   If LOC = -1 then: Write: x is not in the array A.
Else: Write: LOC is the location of x
[End of If structure]
6.   Exit

Complexity: The complexity is $O(\log_2 n)$.

Problem IV: Matrix Multiplication.
Algorithm3.4: MATMUL(A,B,C,M,P,N)
(Let A be an M × P matrix array and B be an P × N matrix array. This algorithm stores the product of A and B in an M×N matrix array C)
1.   Repeat steps 2 to 4 for I = 1 to M:
2.   Repeat steps 3 and 4 for J = 1 to N:
3.   Set C[I,J]:=0.
4.   Repeat for K = 1 to P: C[I,J] :=C[I,J] + A[I,K]*B[K,J].
[End of step 4 loop]
[End of step 2 loop]
[End of step 1 loop]
5.   Exit

Complexity: The complexity is $O(n^3)$.

MORE PROBLEMS

1.   Programming Problems of Chapter 4 of "Data Structures" by Seymour Lipschutz.

LAB REPORT: You have to submit all assigned problems in next lab.