ARRAY TECHNIQUES

classmate
Date 21-9-18
Page

Array is a powerful tool that is widely use in computing arrays provide for a very spacial way of storing or organising data in Computers memory The power of array is Largely derived from the fact that it provides us with very simple & efficient way of ~~reffe~~ referring to and performing Computations on Collection.

- Array Share Some Common attribute:-

The attribute that also called name Can be Share by each element of the array. each element has unique address and which distinguishes from all Other elements.

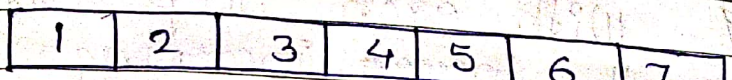
* Arrays are of types :-
  i) 1 Dimensional (Least) array.
  ii) 2 Dimensional (table or matrix) array
  iii) Multy dimensional array.

* Array order Reversal :-

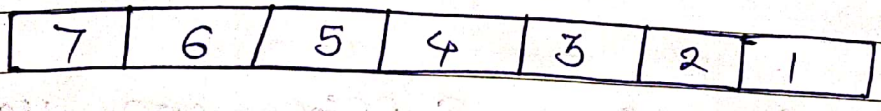
Problem :-

~~Requ~~ Rearranged element in an array so that they appear in reverse order.

* Discription (Developement)



I/P before reversal

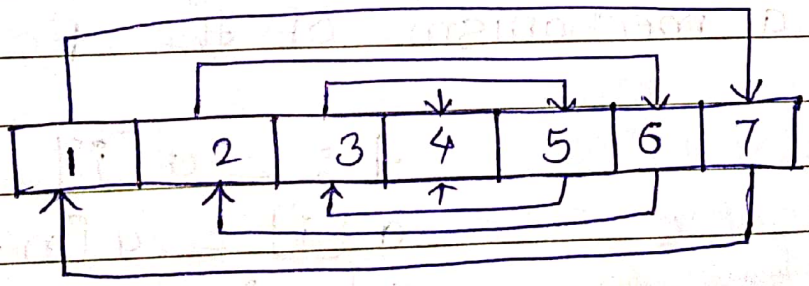


After reversal O/P.

We can start the design of algorithm by Carefull examination of the elements of an array before & after it has been reverse as shown in above.

We observe that the first element ends up in Last position. the second elements ends up in second Last position & so on.

Carring this process we get the following set of exchanges in terms of suffix.



Step [1] a[1] ⟺ a[7]
Step [2] a[2] ⟺ a[6]
Step [3] a[3] ⟺ a[5]
Step [4] a[4] ⟺ a[4] there is no exchange

We observe that after Step 3 the array is completly reverse we see that With each step the Suffix on the Length increasing by 1 & while the Suffix on the right are decreasing by 1.

For increasing Suffix we use Variable [i] which is simplely increamented by 1 with each step.

For our decreasing suffix we might try [n-i] as a suffix since this decreases by 1 with each increase i by 1

The problem with this Solution is that when i=1 we find n-i is equal to n-1 rather than 1.

So we correct this by adding 1 So the decreasing Suffix become n-i+1.

Now, each exchange can be achived by a mechanism of the form

```
t := a[i]
a[i] = a[n-i+1]
a[n-i+1] = t
```

Algorithm discription

Step [1] establish the array a[1....n] of n element to be reverse.

Step [2] Computer r
The no. of exchanges needed to reverse the array. $\left[r = \frac{n}{2}\right]$.

Step [3] While there are still pair of array element to be exchange (while $r > 0$)

a) exchange the i^{th} element $(n-i+1)^{th}$ element.

b) decrease r by 1.

Step [4] Written the reverse array.

* Array Counting or histogramy :-

Problem :-

Given a set of n students examination marks in the range 0 to 100 make a count a no. of student that obtain each possible mark.

Development -

In this problem we distribute a set of mark and create that many Counting variables each corresponding to perticular mark as follows.

a) get next mark.

b) if mark = 0 then $C_0 = C_0 + 1$

b) if m = 1 then $C_1 = C_1 + 1$

b) if m = 2 then $C_2 = C_2 + 1$

.

.

.

if m = 100 then $C_{100} = C_{100} + 1$

difficultly with this approch we need to make 101 test just to update Count For 1 perticular mark. & the program becomes very Long.

For that we consider unother approch which is basically manual & than applied For Computer Solution in this method it is not neccessary to Compair each mark with all possible Possible marks but instated of that the perticular mark needs us directly to the perticular slot that must be updated.

if we store in each array location the Count of the no. of students that obtain that marks we will have the required Solution of the problem.

eg:- 15 students obtain 57, then we will have no. 15 in location 57 & so on...

∴ New Count in the location
= Previous count in location +1.

eg:- a[57] := a[57] + 1

In general, a[m] := a[m] + 1.

therefore over all algorithm is discription

Step 1:- Prompt & read n the no. of marks to be process.

Step 2:- Initialise all elements of the Counting array a[1....100] to zero.

a[1] = 0
a[2] = 0
⋮
a[100] = 0

Step 3 :- While there are still marks to be process repetedly do.

a) read next mark m.
b) add 1 to the Count in location m in the Counting array.

Step 4 :- Write out marks Frequency Count distribution.

22-9-18

* Finding the maximus no. in a set :-

problem :- Find the maximum no in set of n numbers.

development :- First we must know Clear idea of defination of maximum. i.e. the maximum is that no. which is greater than or equal to all other no.s in the set.

This defination occomoded that the maximum may not be unique. it is also implied that the maximum is only define for set of one or more elements.

to start the development Consider For example

1	8	6	5	15	7	19	21	6	13

this list shows that all no.s need to be examine to establish the maximum. & second conclusion is that comparision of the relative magnitude of no. must be made for that we are following procedure.

1) When first no. appeares on the screen we have no way of ~~Ko~~ knowing whether ~~whether~~ or not it is maximum. In this situations the best that we write down this no. as our temporary maximum Candidate.

2) read the second no. three situations are possible the second no. can be less than our temprary maximum no.

2) The second no. can be equal to temparary maximum no.

3) the second no. can be greater than our temporary maximum no. In situations one and ~~two~~ two there is no need to exchange it we simply go ahead and compair the third no. however if second no. is greater than we must exchange the two no.s & the second no. becomes temporary maximum no. this is the process & will need to continue untill all

elements in the set have been examin... & the last values are encounter they assume the role of temparary maximum.

Discription :-

Step [1] = establish an array a [1.....n] of n element where n >= 1.

2) set temparary maximum max to the first array element (max := a [1])

3) While less than n array element have been consider
   a) if next element greater than the current max no. than assign it to max, max := a [2],

4) Written maximum from the array is value of max.

Note :-

For n element n-1 Comparisions needed to find the maximum in an array.

—— o ——