

Frequency Count of Array Element

Indian Institute of Information Technology, Allahabad
ITP Assignment - 1

KRITI SHUKLA

MANYA AGRAWAL

Prof - Dr Mohammed Javed Sir



NAMAN SUHANE

GADAMSETTY
ASWITHA SAI

TA - Md Miraz Sir



CONTENTS

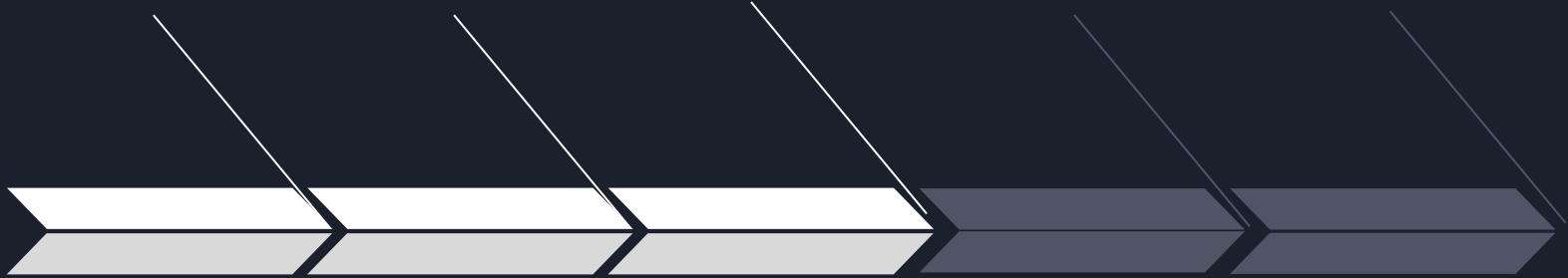
PROBLEM
STATEMENT

ALGORITHM

PSEUDO CODE

TIME COMPLEXITY

CONCLUSION





PROBLEM STATEMENT

Write a C program to count the frequency of each element and print all



ALGORITHM

1. Declare and initialize an array `arr`.
2. Declare another array `fr` with the same size of array `arr`. It is used to store the frequencies of elements present in the array.
3. Variable `visited` will be initialized with the value `-1`. It is required to mark an element visited, which help us to avoid counting the same element again.
4. The frequency of an element can be counted using two loops. One loop will be used to select an element from an array, and another loop will be used to compare the selected element with the rest of the array.
5. Initialize count to 1 in the first loop to maintain a count of each element. Increment its value if a duplicate element is found in the second loop. Since we have counted this element and didn't want to count it again. Mark this element as visited by setting `fr[j]=visited`. Store count of each element to `fr`.
6. Finally print the elements in the array along with their frequencies .



PSEUDO CODE

STEP 1: Start

STEP 2: INITIALIZE

`arr[]={1,2,2,1,3,3,5,3,1}`

STEP 3: `length=sizeof (arr)/sizeof (arr [0])`

STEP 4: DEFINE `fr[length]`.

STEP 5: SET `visited =-1`.

STEP 6: SET `i= 0`. REPEAT STEP 7 to 12 until `i<length`

STEP 7: SET `count =1`

STEP 8: SET `j=0`.REPEAT STEP 9 and 10 until `j<length`.

STEP 9: `If(arr[i]==arr[j])then`

`Count++`

`fr[j]=visited`

STEP 10: `j=j+1`

STEP 11: `if(fr[i]!=visited)`
`thenfr[i]=count`

STEP 12: `i=i+1`

STEP 13: PRINT “-----”

STEP 14: PRINT
“Element|Frequency”

STEP 15: PRINT “-----”

STEP 16: SET `i=0`. REPEAT STEP 17 and 18 until `i<length`

STEP 17: `if(fr[i]! =visited)`

Then print (“frequency of the
“”element is “”)

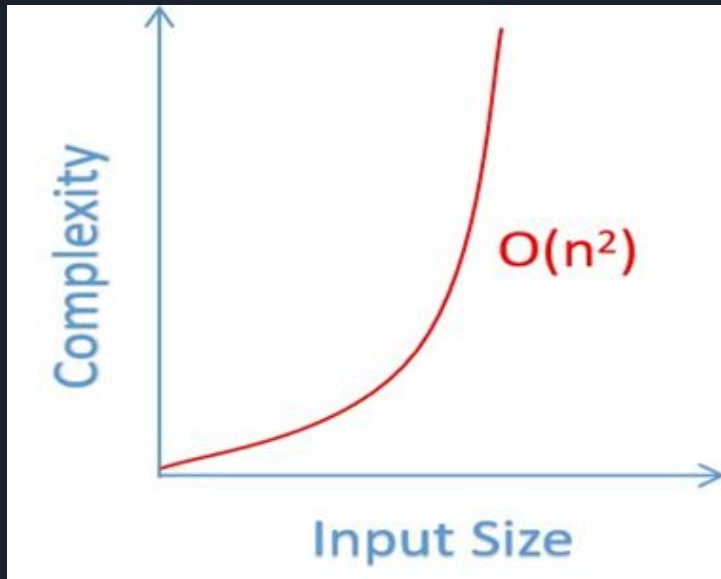
STEP 18: `i= i+1`.

STEP 19: PRINT “-----”

STEP 20: RETURN 0.

STEP 21: END

TIME COMPLEXITY

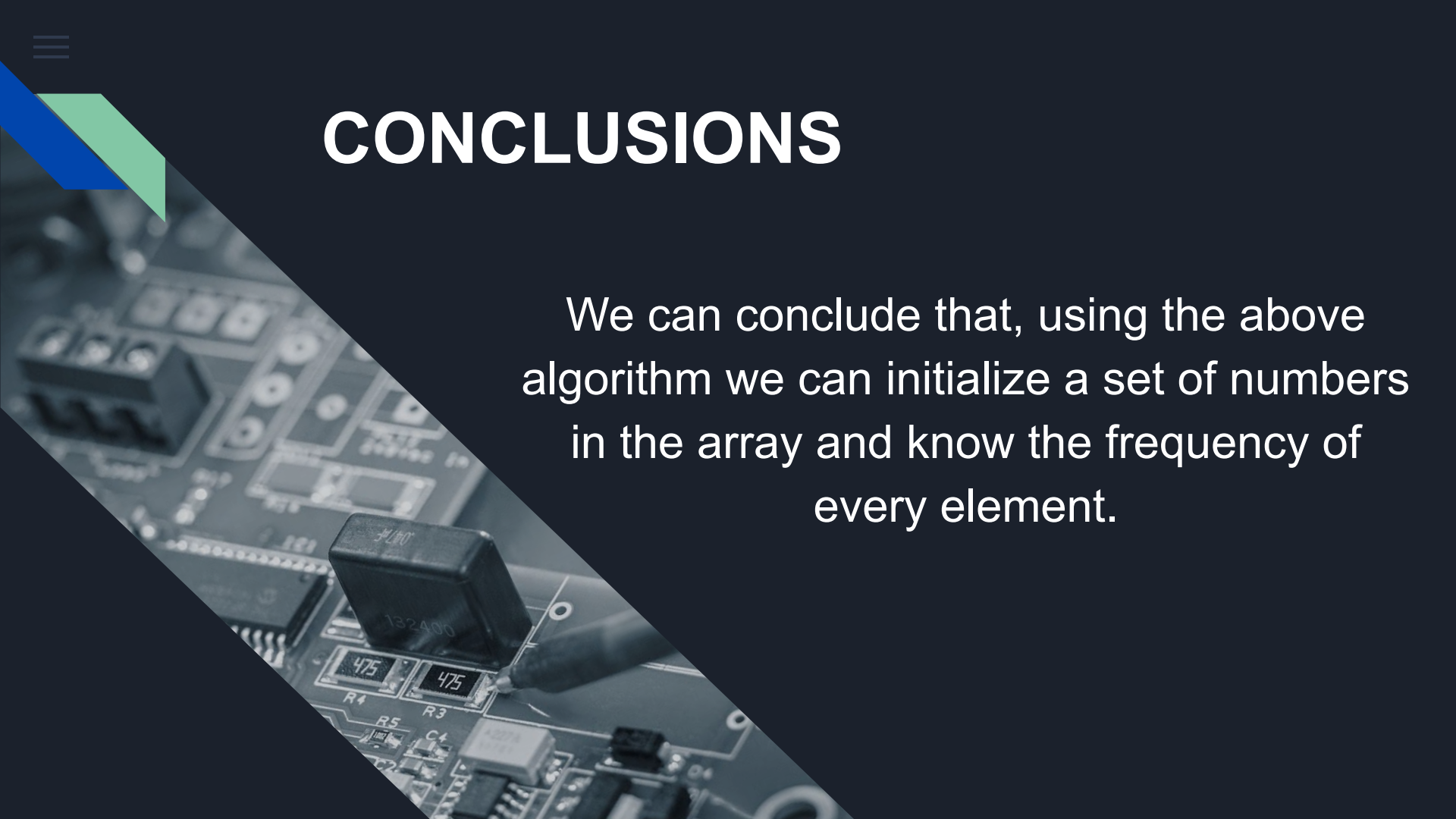


- In the above algorithm to find the frequency of an element we used two nested for loops.
- One loop to select an element from an array, and another to compare the selected element with the rest of the array.
- This resulted in the time complexity of $O(n^2)$.



CONCLUSIONS

We can conclude that, using the above algorithm we can initialize a set of numbers in the array and know the frequency of every element.

A detailed background image of a printed circuit board (PCB) is visible, showing various electronic components such as resistors, capacitors, and integrated circuits. A soldering iron is shown in the process of soldering a component onto the board. The image is partially obscured by a dark blue diagonal overlay that contains the text.