Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

Department of
Artificial Intelligence and Data Science

Name: Siddhesh Dilip Khairnar		
Class: SY	Division: B	Roll No: 272028
Semester: III	Academic Year: 2022-2023	
Subject Name & Code: Data Structure, ADUA21202		
Title of Assignment: Sort the data in ascending order using Bubble sort (Display pass by pass output) and search a particular data using Binary search.		

Assignment No.- 9

Ds Assignment - 9

Name: Siddhesh Dilip Khairnar
Rollno: 272028   PANNO: 22110398

Aim: sorting the data in ascending order.

Problem statement: Sort the data in ascending Order using Bubble sort (Display Pass by Pass Output) and search a Particular data using Binary search.

Background: →

- Bubble sort is the simplest sorting algorithm that work by repeatedly swapping the adjacent element if they are in the wrong order. This algorithm is not suitable for large data set as its average and worst-case time complexity is quite high. It time complexity is $O(N^2)$ and Auxiliary space is $O(1)$.

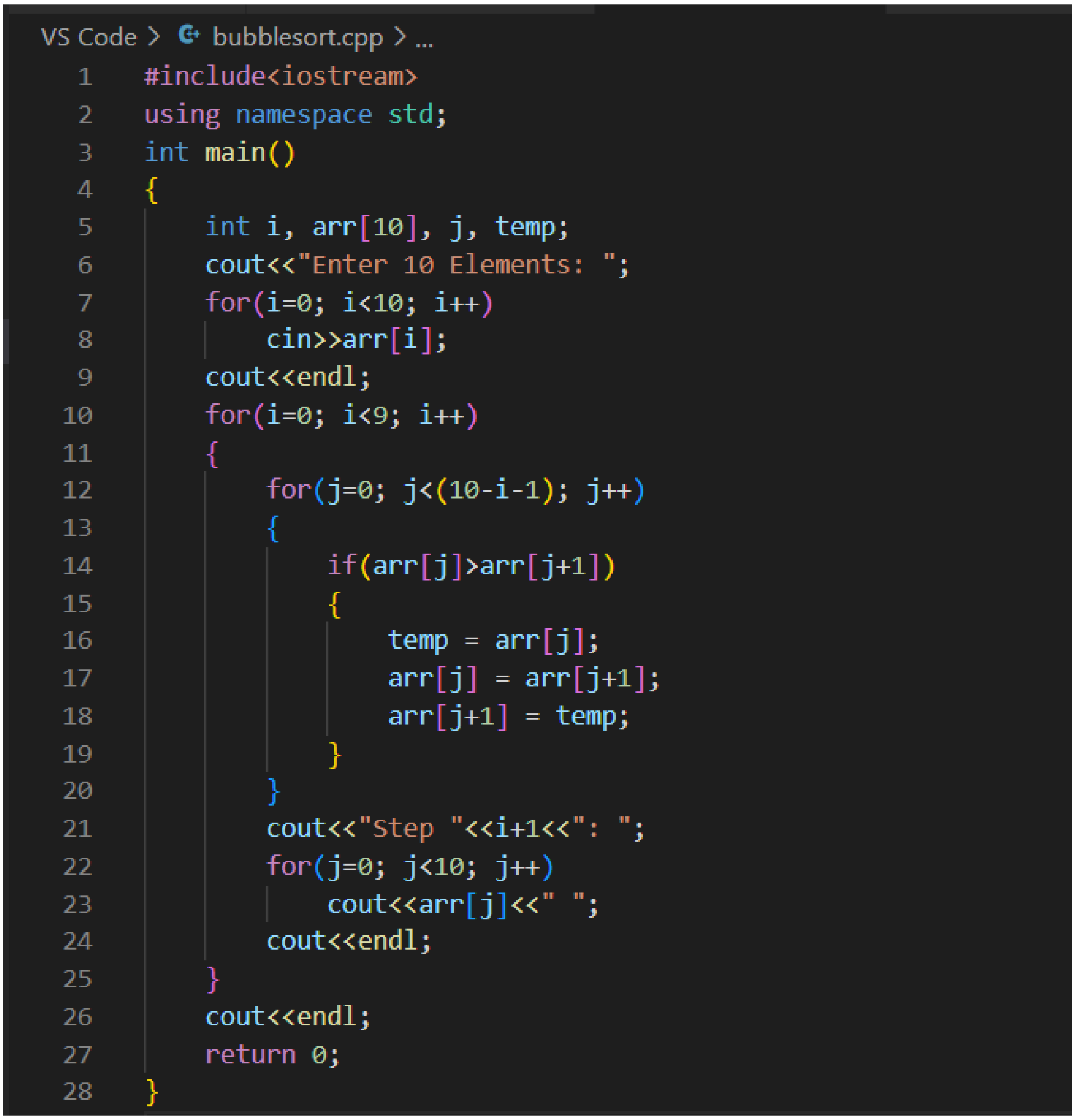
- Binary search is an efficient algorithm for finding an item from a sorted list of item. It work by repeatly dividing in half the portion of the list that could contain the item, ~~untill~~ until you've narrowed down the Possible location to just one.

Software requirement: online compiler Or Any IDE.

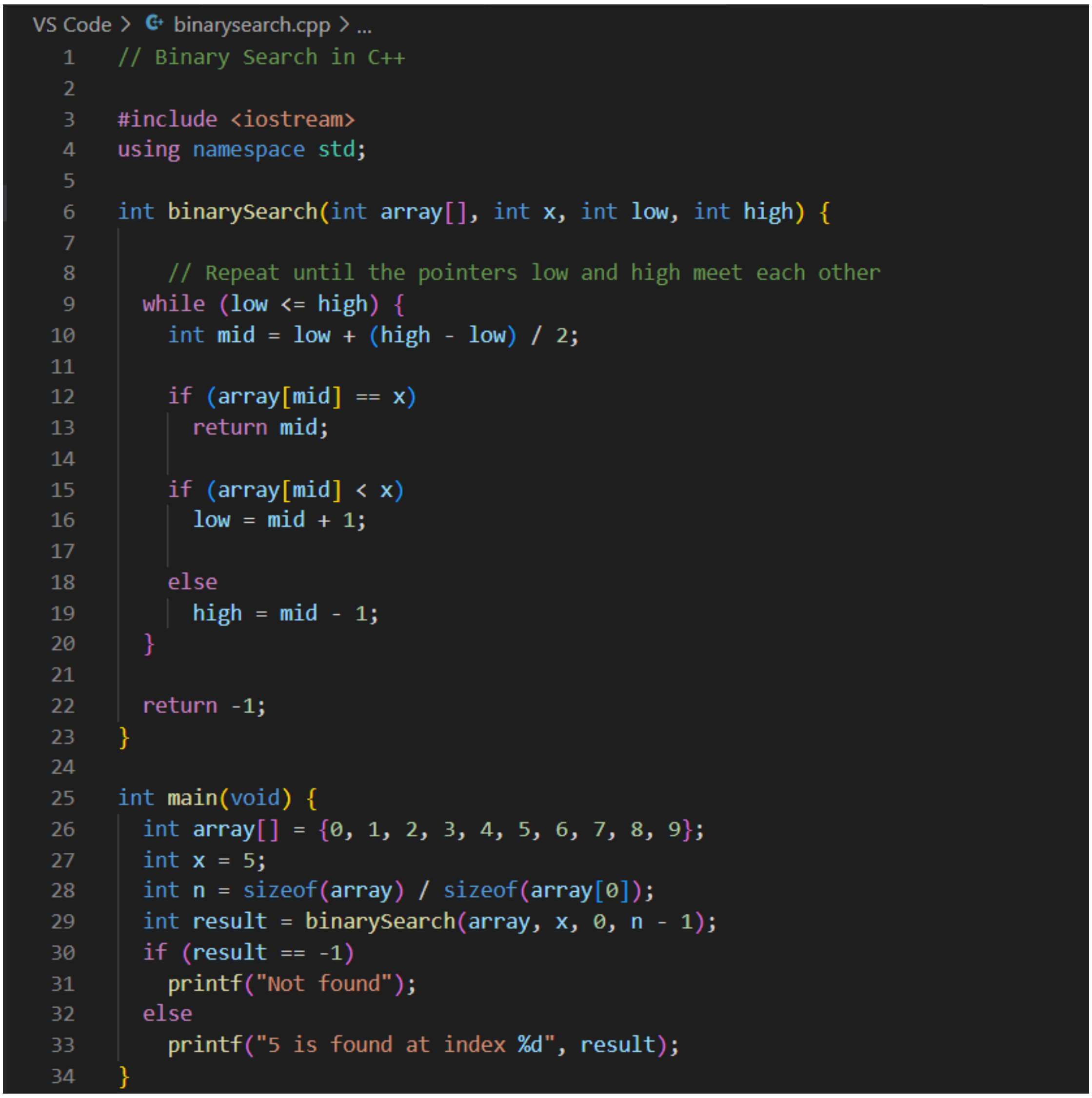
Conclusion: Learn the sorting of Data using ~~Bubble~~ Bubble Sort and Binary search. Also Learn the some the Basic different ~~betweet~~ between them.

Ashwin

Program:

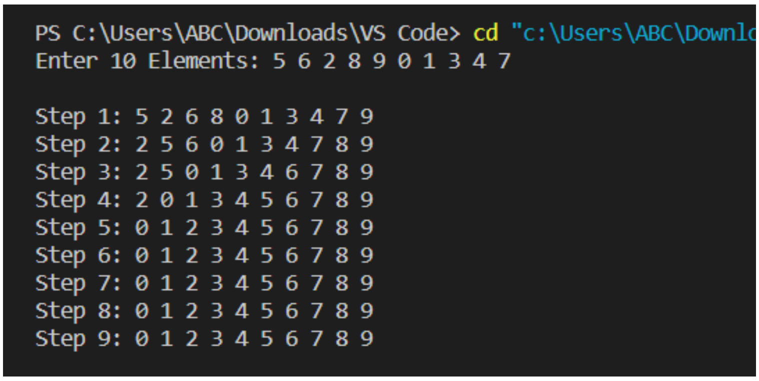


```cpp
#include<iostream>
using namespace std;
int main()
{
    int i, arr[10], j, temp;
    cout<<"Enter 10 Elements: ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<endl;
    for(i=0; i<9; i++)
    {
        for(j=0; j<(10-i-1); j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
        cout<<"Step "<<i+1<<": ";
        for(j=0; j<10; j++)
            cout<<arr[j]<<" ";
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

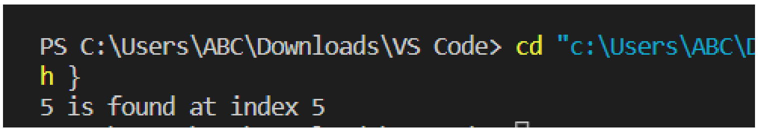


```cpp
// Binary Search in C++

#include <iostream>
using namespace std;

int binarySearch(int array[], int x, int low, int high) {

    // Repeat until the pointers low and high meet each other
  while (low <= high) {
    int mid = low + (high - low) / 2;

    if (array[mid] == x)
      return mid;

    if (array[mid] < x)
      low = mid + 1;

    else
      high = mid - 1;
  }

  return -1;
}

int main(void) {
  int array[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
  int x = 5;
  int n = sizeof(array) / sizeof(array[0]);
  int result = binarySearch(array, x, 0, n - 1);
  if (result == -1)
    printf("Not found");
  else
    printf("5 is found at index %d", result);
}
```

Output:



```
PS C:\Users\ABC\Downloads\VS Code> cd "c:\Users\ABC\Downlo
Enter 10 Elements: 5 6 2 8 9 0 1 3 4 7

Step 1: 5 2 6 8 0 1 3 4 7 9
Step 2: 2 5 6 0 1 3 4 7 8 9
Step 3: 2 5 0 1 3 4 6 7 8 9
Step 4: 2 0 1 3 4 5 6 7 8 9
Step 5: 0 1 2 3 4 5 6 7 8 9
Step 6: 0 1 2 3 4 5 6 7 8 9
Step 7: 0 1 2 3 4 5 6 7 8 9
Step 8: 0 1 2 3 4 5 6 7 8 9
Step 9: 0 1 2 3 4 5 6 7 8 9
```



```
PS C:\Users\ABC\Downloads\VS Code> cd "c:\Users\ABC\D
h }
5 is found at index 5
```