[closed] Data Structure Tutorial : Array

5

```
If you find any error please comment. I will try to update this post.
Array is a collection of homogeneous data elements. It is a very simple data
structure. The elements of an array are stored in successive memory location.
Array is refered by a name and index number. Array is nice because of their
simplicity and well suited for situations where the number is known. Array
operation :

Traverse
Insert
 Delete
Sort
Search


There are two types of array. One dimensional array and multi dimensional
array.One dimensional array This type of array of array represent and strore in
linear form. Array index start with zero.

Declaration : datatype arrayname[size];
                        int arr[10];

Input array : for(int i=0; i<10; i++)  cin>>arr[i];

We can use store integer type of data to the array arr using above segment.

----------

Traverse : Traversing can easy in linera array. Algorithm:


C++ implement :
void traverse(int arr[])
{
       for(int i=0; i<10; i++)   cout<<arr[i];
}

----------

Insertion : Inserting an element at the end of a linear array can be easily done
provided the memory space space allocated for the array is large enough to
accommodate the additional element. Inserting an element in the middle . .
Algorithm : Insertion(arr[], n, k, item) here arr is a linear array with n
elements and k is index we item insert. This algorithm inserts an item to kth in
index in arr.

Step 1:Start
Step 2: Repeat for i=n-1 down to k(index)
   Shift the element dawn by one position] arr[i+1]=arr[i];
  [End of the loop]
Step 3: set arr[k] = item
Step 4: n++; Step 5 : Exit.


C++ implement :
void insert(int arr[], int n, int k, int item)
{
   for(int i=n-1; i>=k; i--)
      {
         arr[i]=arr[i+1];
      }
      arr[k] = item;
      n++;
}

----------

Deletion : Deletion is very easy on linear array.

Algorithm : Deletion(arr, n, k) Here arr is a linear array with n number of
```

```
items. K is position of elememt which can be deleted.
Step 1:Start
Step 2: Repeat for i=k upto n
       [Move the element upword]  arr[i]=arr[i+1];
      [End of the loop]
Step 3: n--;
Step 4 : Exit.


C++ implementation :
 void deletion(int arr[], int n, int k)
 {
      for(int i=k; i<n; i++)
       {
            arr[i] = arr[i+1];
       }
      n--;
  }

----------

Searching : Searching means find out a particular element in linear array.
Linear seach and binary search are common algorithm for linear array. We discuss
linear search and binary search.

Linear search Algorithm : Linear search is a simple search algorithm that checks
every record until it finds the target value
Algorithm: LinearSeach(arr, n, item)
Step 1:Start.
Step 2: Initialize loc=0;
Step 3: Repeat for i=0 upto n-1 if(arr[i]==item) loc++; [End of the loop]
Step 4: if loc is not zero then print found otherwise print not found.
Step 5 : Exit.

C++ implementation :
  void linear search(int arr[], int n, item)
   {
      for(int i=0; i<n-1; i++)
       {
         if(arr[i]==item) loc++;
       }

      if(loc) cout<<"Found"<<endl;
      else cout<<"Not found"<<endl
   }

----------

Binary search : Binary search is available for sorted array. It compares the
target value to the middle element of the array; if they are unequal, the half
in which the target cannot lie is eliminated and the search continues on the
remaining half until it is successful.

Algorithm : BinarySeach(arr, n, item)
Step 1:Start
Step 2: Initialize low = 0 and high = n-1;
Step 3: While loop low<=high
    mid = (low + high)/2;
    if (a[mid] == item) return mid;
    else if (a[mid] < item) low = mid + 1;
     else high = mid - 1;
Step 4: If item is not found in array return -1. Step 5: End.


C++ implementation :
 int binarySearch(int[] a, int n, int item)
 {
    int low = 0;
    int high = n - 1;
    while(low<=high){
    {
      int mid = (low + high)/2;
      if (a[mid] == item) return mid;
      else if (a[mid] < item) low = mid + 1;
      else high = mid - 1;
    }
    return -1;
 }


--------------------
Sorting : There are various sorting algorithm in linear array. We discuss bubble
sort and quick sort in this post.
Bubble Sort: Bubble sort is a example of sorting algorithm . In this method we
at first compare the data element in the first position with the second position
and arrange them in desired order.Then we compare the data element with with
third data element and arrange them in desired order. The same process
```

```
continuous until the data element at second last and last position.

Algorithm : BubbleSort(arr,n)
Step 1:Start
Step 2: Repeats i=0 to n
Step 3: Repeats j=0 to n
         if(arr[j]>arr[j+1]) then interchange arr[j] and arr[j+1]
         [End of inner loop]
        [End of outer loop]
Step 4: Exit.

C++ implement :
 void BubbleSort(int arr, int n)
 {
    for(int i=0; i<n-1; i++)
    {
        for(int j=0; j<n-1; j++)
        {
           if(arr[j]>arr[j+1])     swap(arr[j],arr[j+1]);
        }
     }
 }

Quick Sort:
 Quick sort is a divide and conquer paradism. In this paradism one element is to
be chosen as partitining element .
We divide the whole list array into two parts with respect to the partitiong
elemnt . The data which are similar than or equal to the partitining element
remain in
 the first part and data data which are greater than the partitioning element
 remain in the second part. If we find any data which is greater than the
partitioning value that will be transfered to the second part., If we find any
data whichis smaller than the partitioning element that will be transferred to
first part.
Transferring the data have been done by exchanging the position of the the data
found in first and second part. By repeating this process ,
we can sort the whole list of data.

Algorithm: QUICKSORT(arr, l, h)
if  l<h then pi ← PARTITION(A, l, h)
QUICKSORT(A, l, pi–1)
QUICKSORT(A, pi+1, h)

C++ implementation :

 int partition(int arr[], int start, int end)
 {
    int pivotValue = arr[start];
    int pivotPosition = start;
     for (int i=start+1; i<=end; i++)
     {
        if (pivotValue > arr[i])
       {
          swap(arr[pivotPosition+1], arr[i]);
          swap(arr[pivotPosition] , arr[pivotPosition+1]);
          pivotPosition++;
       }
     }
    return pivotPosition;
  }


 void quickSort(int arr[], int low, int high)
 {
   if (low < high)
   {
      int pi = partition(arr, low, high);
      quickSort(arr, low, pi - 1);
      quickSort(arr, pi + 1, high);
   }
 }





C++ example for simple sorting program with stl function:

 #include <bits/stdc++.h>
 using namespace std;

 int main()
 {
   int  n, arr[100];
   cin >> n;
   for(int i=0; i<n; i++)
```

```
    {
      cin>>arr[i];
    }

    sort(arr, arr + n);

    for (int i=0; i<n; i++)
    {
       cout<<arr[i]<<" ";
    }
    cout<<endl;

    return 0;
  }
```

Codechef Problem : SMPAIR, Ups and Downs, KTTABLE, TLG ,FORESTGA
Spoj Problem : AGGRCOW - Aggressive cows
Hackerrank Problem : Arrays - DS, Quicksort 1 - Partition, Quicksort 2 - Sorting

datastructure

closed **01 Jan '17, 11:45**

asked **20 Nov '16, 22:26**
1★ rashedcs
[497]●1●7●33
accept rate: 4%

The question has been closed for the following reason "Other" by rashedcs 01 Jan '17, 11:45

8 Answers: oldest answers newest answers **popular answers**

2

For the deletion algo, wouldn't k-1 be the position of element to be deleted?

link | award points

answered **21 Nov '16, 11:00**
byteword
[26]
accept rate: 100%

I modified this post. Tnq for finding error.

1★ rashedcs (21 Nov '16, 11:35)

2

Nice tutorial. Upvoted.

link | award points

answered **21 Nov '16, 11:53**
5★ mathecodician
[2.6k]●4●23
accept rate: 8%

0

How do we do a binary search in a matrix or a 2-D array whose columns and rows are sorted? And how to sort a 2-D array?

link | award points

edited **21 Nov '16, 11:52**

answered **21 Nov '16, 11:52**
5★ mathecodician
[2.6k]●4●23
accept rate: 8%

2 https://www.quora.com/How-do-I-search-in-a-row-wise-and-column-wise-sorted-matrix

1★ rashedcs (21 Nov '16, 11:56)

Thanks and how do we sort a 2-D array?

5★ mathecodician (21 Nov '16, 11:58)

0

@rashedcs types of array: one-dimensional and two-dimensional, shouldn't it be multi-dimensional array instead of two-dimensional array, as in dynamic programming sometimes 3D and 4D arrays are also used.

link | award points

edited **21 Nov '16, 13:23**

answered **21 Nov '16, 11:59**
5★ srd091
[1.5k]●1●11
accept rate: 42%

0

In quick sort, you have written "here's the link". The link is not working.

link | award points

answered **21 Nov '16, 12:01**
5★ mathecodician
[2.6k]●4●23
accept rate: 8%

where you have written 'details about quick sort[link]', the link is still not working.

5★ mathecodician (21 Nov '16, 13:25)

2 Link : http://quiz.geeksforgeeks.org/quick-sort/

1★ rashedcs (22 Nov '16, 13:29)

0

@rashedcs Small typo its bubble sort not buble sort or bible sort ...

Some suggestions to improve this tutorial , try to explain each function in sorting separately , like provide quicksort(parameters) and partition(parameters) example separately with examples as raw theory is hard to understand for someone who is coming across these algo for first time. Also when you use any built in function try to include its other overloaded versions if any.

link | award points

answered **21 Nov '16, 12:02**
3★ smsubham
[673]●2●14
accept rate: 15%

Yeap. I always try to improve my tutorial. Tnq for suggestion @smsubham

1★ rashedcs (21 Nov '16, 12:09)

2 It is typing mistake. I modified that.

1★ rashedcs (21 Nov '16, 13:20)

How do we sort a 2-D array by rows and columns both?

0 link | award points

answered **22 Nov '16, 13:05**
5★ mathecodician
[2.6k]●4●23
accept rate: 8%

Click this link: http://stackoverflow.com/questions/33063509/how-to-sort-a-2d-array-row-and-column-wise

1★ rashedcs (22 Nov '16, 13:21)

You should add the tag algorithm as this tutorial is more about algo's I think.

0 link | award points

answered **23 Nov '16, 15:06**
5★ mathecodician
[2.6k]●4●23
accept rate: 8%

yeap. You are right.

1★ rashedcs (23 Nov '16, 15:09)