

cprogramto.com

My Project - Video Waker Alarm



video waker alarm

C PROGRAM

C PROGRAM FOR QUICK SORT

JUNE 10, 2014 | AKSHAY UPADHYAY

C program for Quick Sort. Quick Sort also known as **Partition exchange** sort, is a comparison **unstable -in-place sorting algorithm**. Quick Sort compare n items in **$O(n \log n)$** time complexity in average and best case, while takes **$O(n^2)$ in worst case**. It comes under divide and conquer algorithm which divides the large set of array in small by picking up a pivot element as a reference element, and do sorting. You can also see other **sorting algorithm implementation** like, **Heap sort**, **bubble sort**, **Insertion sort**, **selection sort** etc.



c program for quick sort

Quick Sort Algorithm Steps:

- 1.) Make a list of items that need to be sorted, lets apply in an array.
- 2.) Choose any element as **pivot** element from the array list.(Complexity largely depends on choosing the pivot element)

3.) Rearrange the array list so that all the elements with **value less than the pivot will come before the pivot and the element with value greater will come after the pivot with in the same array**, which make pivot element in the sorted position.(If the reverse the order we are reversing the sorting order that is descending).

4.) Apply **recursively the 3rd step** to the sub array of the element with smaller values and separate the sub array of the elements with the greater values.

Note: In the below implemented quick sort program, always first element is set as Pivot element. But if array is already sorted, then the time complexity becomes worse, as quick sort will take quadratic time. So pivot element should be chosen in such a way, that partitioning of array becomes half.

C program for Quick Sort

```
#include<stdio.h>
#include<conio.h>

//quick Sort function to Sort Integer array list
void quicksort(int array[], int firstIndex, int lastIndex)
{
    //declaaring index variables
    int pivotIndex, temp, index1, index2;

    if(firstIndex < lastIndex)
    {
        //assigninh first element index as pivot element
        pivotIndex = firstIndex;
        index1 = firstIndex;
        index2 = lastIndex;

        //Sorting in Ascending order with quick sort
        while(index1 < index2)
        {
            while(array[index1] <= array[pivotIndex] && index1 < lastIndex)
            {
```

```
        index1++;
    }
    while(array[index2]>array[pivotIndex])
    {
        index2--;
    }

    if(index1<index2)
    {
        //Swapping operation
        temp = array[index1];
        array[index1] = array[index2];
        array[index2] = temp;
    }
}

//At the end of first iteration, swap pivot element with index2 element
temp = array[pivotIndex];
array[pivotIndex] = array[index2];
array[index2] = temp;

//Recursive call for quick sort, with partitioning
quicksort(array, firstIndex, index2-1);
quicksort(array, index2+1, lastIndex);
}
}

int main()
{
    //Declaring variables
    int array[100],n,i;

    //Number of elements in array form user input
    printf("Enter the number of element you want to Sort : ");
    scanf("%d",&n);

    //code to ask to enter elements from user equal to n
    printf("Enter Elements in the list : ");
```

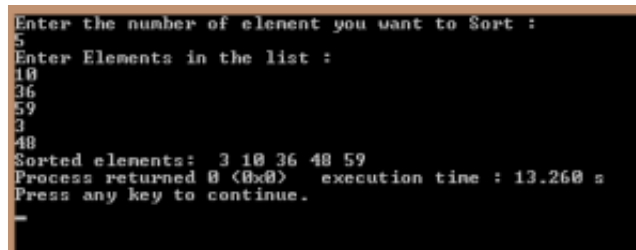
```
for(i = 0; i < n; i++)
{
    scanf("%d",&array[i]);
}

//calling quickSort function defined above
quicksort(array,0,n-1);

//print sorted array
printf("Sorted elements: ");
for(i=0;i<n;i++)
printf(" %d",array[i]);

getch();
return 0;
}
```

Output of C program for Quick Sort



```
Enter the number of element you want to Sort :
5
Enter Elements in the list :
10
36
59
3
48
Sorted elements: 3 10 36 48 59
Process returned 0 (0x0)   execution time : 13.260 s
Press any key to continue.
=
```

c program for quick sort

Related Posts via Categories

- [C program for Selection Sort](#)
- [C program for Insertion Sort](#)
- [C program to convert Binary to float](#)
- [C program to convert float to binary](#)
- [C program to sort 2D array by row or column](#)
- [String.h – string functions in c](#)
- [C program to sort 2D array](#)
- [C program for ASCII number from ASCII code](#)

- C program for ASCII code of number
- ASCII code print c program

◀ QUICK SORT ◀ SORT ◀ SORTING PROGRAM