

C Programming

C Programming | Networking | General Computer | Exam Questions & Answers | How To Article | Tips & Tricks | PHP Programming | Source Code | PHP | MySQL | Pointer | Operators | Constant | Variable | Keyboard | Algorithm | Flowchart | Storage Classes | Function | Your Comment Appreciate Me.

Home	Pyramid	C Tutorial	Questions/Answers	C Quiz	Networking	PHP	
------	---------	------------	-------------------	--------	------------	-----	--

g+1 1.7k

2.10.2012

Quick sorting

Quick sort is a divide and conquer algorithm. Its divided large list in mainly three parts:

1. Elements less than pivot element.
2. Pivot element.
3. Elements greater than pivot element.

Where pivot as middle element of large list. Let's understand through example:

List : 3 7 8 5 2 1 9 5 4

In above list assume **4** is pivot element so rewrite list as:

3 1 2 4 5 8 9 5 7

Here, I want to say that we set the **pivot element(4)** which has in left side elements are less than and right hand side elements are greater than. Now you think, how's arrange the less than and greater than elements? Be patient, you get answer soon.

Now let's start understand the concept of quick sort. The **steps are:**

1. Pick a pivot element.
2. Reorder the list so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it (equal values can go either way). After this partitioning, the pivot is in its final position. This is called the **partition** operation.
3. **Recursively sort** the sub-list of lesser elements and the sub-list of greater elements.

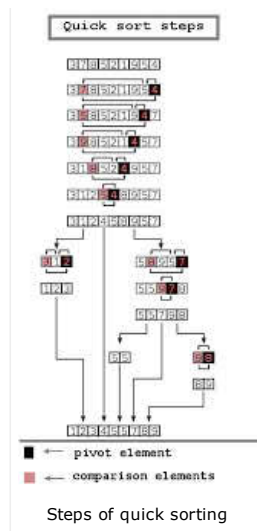
The base case of the recursion are lists of size zero or one, which never need to be sorted

Example of quick sort process:

Search Programs

C Programs INDEX

- ▶ [2015](#) (5)
- ▶ [2014](#) (36)
- ▶ [2013](#) (84)
- ▼ [2012](#) (156)
 - ▶ [December](#) (12)
 - ▶ [November](#) (11)
 - ▶ [October](#) (23)
 - ▶ [September](#) (20)
 - ▶ [August](#) (18)
 - ▶ [July](#) (13)
 - ▶ [June](#) (3)
 - ▶ [May](#) (1)
 - ▶ [April](#) (5)
 - ▶ [March](#) (9)
 - ▼ [February](#) (27)
 - [typedef datatype](#)
 - [Heap sorting](#)
 - [Number rhombus2](#)
 - [Merge sorting](#)
 - [Symbol rhombus](#)
 - [Number rhombus](#)
 - [Character rhombus](#)
 - [Square triangle4](#)
 - [Square triangle3](#)
 - [Square triangle2](#)
 - [Square triangle1](#)
 - [Insertion sorting using function](#)
 - [Insertion sorting](#)
 - [Quick sorting](#)
 - [Shell sorting](#)
 - [Comparison storage classes](#)
 - [External storage class](#)
 - [Static Storage Class](#)
 - [Register Storage Class](#)



```

/*c program for quick sorting*/
#include<stdio.h>
#include<conio.h>
void qsort(int arr[20], int fst, int last);
int main()
{
    int arr[30];
    int i,size;
    printf("Enter total no. of the elements : ");
    scanf("%d",&size);
    printf("Enter total %d elements : \n",size);
    for(i=0; i<size; i++)
        scanf("%d",&arr[i]);
    qsort(arr,0,size-1);
    printf("Quick sorted elements are as : \n");
    for(i=0; i<size; i++)
        printf("%d\t",arr[i]);
    getch();
    return 0;
}

void qsort(int arr[20], int fst, int last)
{
    int i,j,pivot,tmp;
    if(fst<last)
    {
        pivot=fst;
        i=fst;
        j=last;
        while(i<j)
        {
            while(arr[i]<=arr[pivot] && i<last)
                i++;
            while(arr[j]>arr[pivot])
                j--;
        }
    }
}

```

[Automatic Storage Class](#)
[Storage classes](#)
[String triangle macro](#)
[Character structure](#)
[Number triangle equal](#)
[Character triangle equal](#)
[Character triangle](#)
[Pascal Triangle](#)

► January (14)

► 2011 (101)

Popular Programs

[Algorithms and Flowchart](#)
[Merge sorting](#)
[Pascal Triangle](#)
[Print prime number 1 to 100](#)
[Factorial C program,Algorithm,Flowchart](#)
[Flowchart for prime number](#)
[Heap sorting](#)
[Quick sorting](#)
[Shell sorting](#)
[call by reference swap program](#)

Total Pageviews



3 2 8 6 8 5 9



Feed - RSS / Atom

Subscribe by Email

Email address...

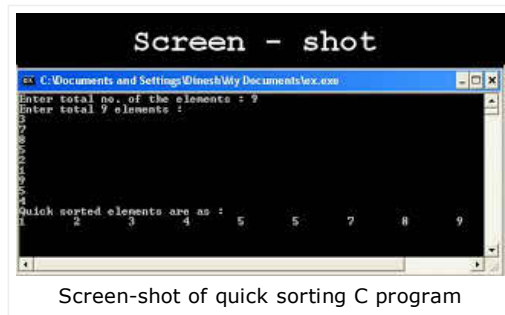
9 Online Now

```

    if (i < j)
    {
        tmp = arr[i];
        arr[i] = arr[j];
        arr[j] = tmp;
    }
}
tmp = arr[pivot];
arr[pivot] = arr[j];
arr[j] = tmp;
qsort(arr, fst, j-1);
qsort(arr, j+1, last);
}
}

```

***** OUTPUT *****



Related programs:

1. [Heap sorting method and algorithm](#)
2. [Heap sorting](#)
3. [Bubble sorting](#)
4. [Selection Sorting](#)
5. [Insertion sorting](#)
6. [Insertion sorting using function](#)
7. [Shell sorting](#)
8. [Merge sorting](#)
9. [Radix sorting](#)
10. [Liner sorting](#)

Posted by [Dinesh Bera](#)

+4 Recommend this on Google

Labels: [example of quick sorting in c](#), [explanation of quick sorting in c](#), [quick sorting in c](#), [quick sorting steps](#)

1 comment:



Jayesh Panchal 27 February 2013 at 20:43

Check out this version of QuickSort Algorithm :)

[Reply](#)

Join this site

with Google Friend Connect

Members (168) [More »](#)



Already a member? [Sign in](#)

SUPPORT C PROGRAMMING



C Programming Langu...
3,981 likes



Be the first of your friends to like this



Live Traffic Feed

A visitor from Philippines

viewed "[C Programming:](#)

[Algorithms and Flowchart](#)" 5

mins ago

A visitor from Bangladesh

viewed "[C Programming: Quick](#)

[sorting](#)" 6 mins ago

A visitor from Pune,

Maharashtra viewed "[C](#)

[Programming: Factorial C](#)

[program Algorithm Flowchart](#)"

A visitor from Coimbatore,

Tamil Nadu viewed "[C](#)

[Programming: Convert](#)

[Fahrenheit to Celsius](#)" 9 mins

ago

A visitor from Pune,

Maharashtra viewed "[C](#)

[Programming: Algorithms and](#)

[Flowchart](#)" 9 mins ago

A visitor from Harare,

Mashonaland East viewed "[C](#)

[Programming: 5.Design triangle](#)

[pyramid](#)" 10 mins ago

A visitor from India viewed "[C](#)

[Programming: Flowchart for](#)

[Fibonacci Series](#)" 11 mins ago

A visitor from Jamnagar,

Gujarat viewed "[C](#)

[Programming: Algorithms and](#)

[Flowchart](#)" 11 mins ago

Mashonaland East viewed "[C](#)

[Programming: Pyramid](#)" 15

mins ago

A visitor from India viewed "[C](#)

[Programming: Increment and](#)

[Decrement operators](#)" 15 mins

[Real-time view](#) · [Get FeedIt](#)

Enter your comment...

Comment as: Google Account

Publish

Preview

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

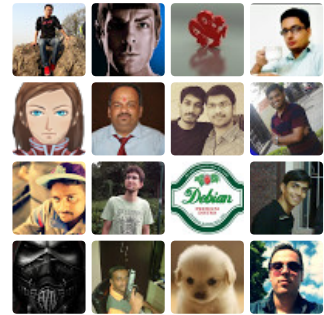
[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Google+ Followers

C Programming

Follow



1,233 have us in circles

[View all](#)

- [Google Tips](#)

Programs Snap Shot

