

Merge Sort

March 15, 2013

MergeSort is a **Divide and Conquer** algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. **The merge() function** is used for merging two halves. The merge(arr, l, m, r) is key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one. See following C implementation for details.

```
MergeSort(arr[], l, r)
```

```
If r > l
```

1. Find the middle point to divide the array into two halves:
middle m = (l+r)/2
2. Call mergeSort for first half:
Call mergeSort(arr, l, m)
3. Call mergeSort for second half:
Call mergeSort(arr, m+1, r)
4. Merge the two halves sorted in step 2 and 3:
Call merge(arr, l, m, r)

The following diagram from [wikipedia](#) shows the complete merge sort process for an example array {38, 27, 43, 3, 9, 82, 10}. If we take a closer look at the diagram, we can see that the array is recursively divided in two halves till the size becomes 1. Once the size becomes 1, the merge processes comes into action and starts merging arrays back till the complete array is merged.



GeeksQuiz



4,126 people like [GeeksQuiz](#).



 Facebook social plugin

ITT Tech - Official Site

[itt-tech.edu](#)

Tech-Oriented Degree Programs.
Education for the Future.



Categories

Articles (28)

C (4)

C++ (2)

Data Structures (13)

DBMS (1)

Operating Systems (1)

Searching and Sorting (7)

Programs (7)

Quizzes (1,392)

Aptitude (1)

Computer Science Quizzes (1,391)

Algorithms (146)

C (203)

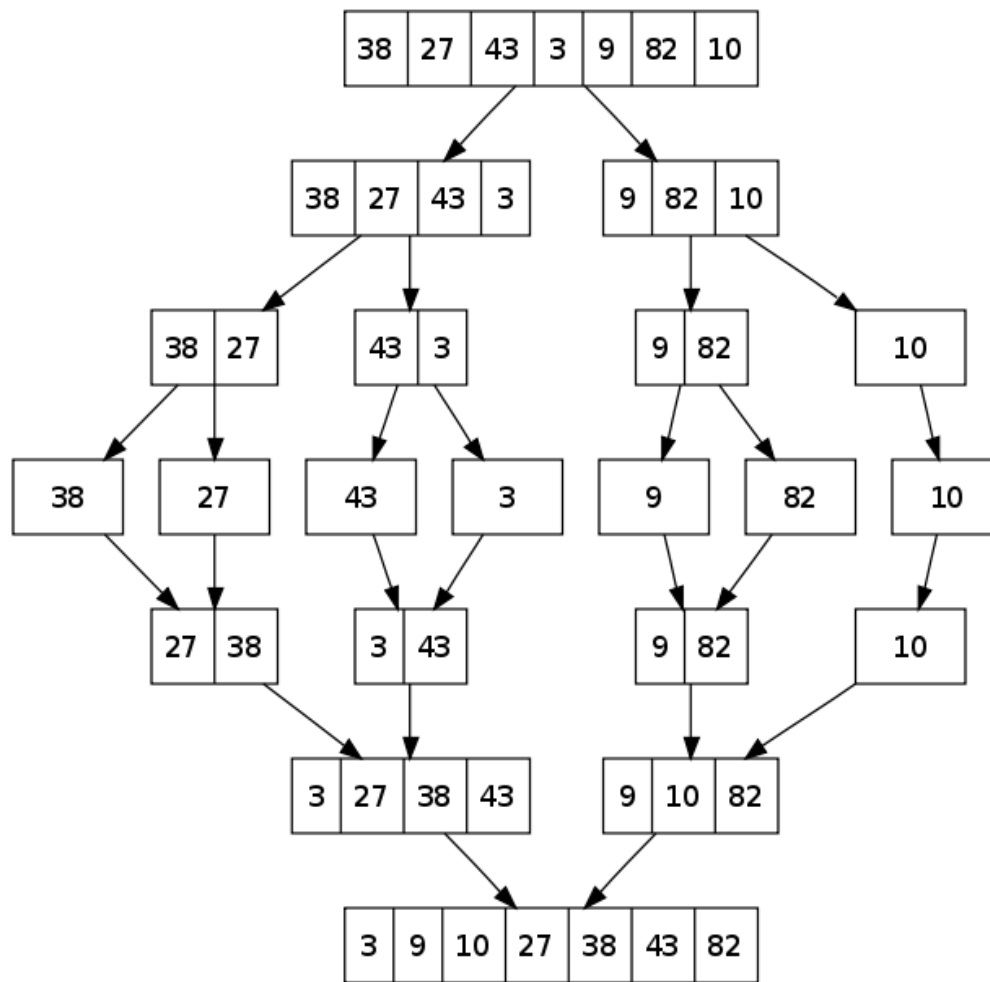
C++ (123)

Data Structures (131)

GATE (709)

Java (51)

Operating Systems (28)



```
/* C program for merge sort */
```

```
#include<stdlib.h>
```

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

```
/* Function to merge the two halves arr[l..m] and arr[m+1..r] of array arr[] */
```

```
void merge(int arr[], int l, int m, int r)
```

```
{
```

```
    int i, j, k;
```

```
    int n1 = m - l + 1;
```

```
    int n2 = r - m;
```

```
    /* create temp arrays */
```

```
    int L[n1], R[n2];
```

```
    /* Copy data to temp arrays L[] and R[] */
```

```

for(i = 0; i < n1; i++)
    L[i] = arr[l + i];
for(j = 0; j < n2; j++)
    R[j] = arr[m + 1 + j];

/* Merge the temp arrays back into arr[l..r]*/
i = 0;
j = 0;
k = l;
while (i < n1 && j < n2)
{
    if (L[i] <= R[j])
    {
        arr[k] = L[i];
        i++;
    }
    else
    {
        arr[k] = R[j];
        j++;
    }
    k++;
}

/* Copy the remaining elements of L[], if there are any */
while (i < n1)
{
    arr[k] = L[i];
    i++;
    k++;
}

/* Copy the remaining elements of R[], if there are any */
while (j < n2)
{
    arr[k] = R[j];
    j++;
    k++;
}
}

/* l is for left index and r is right index of the sub-array
of arr to be sorted */
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {

```

Recent Discussions

Sumit Khatri this is the sorting technique which can work...

Insertion Sort · 7 hours ago

Sumit Khatri no, quick sort requires more swaps than...

Selection Sort · 7 hours ago

Sumit Khatri yes, it is the only sorting technique which...

Selection Sort · 7 hours ago

```

        int m = l+(r-1)/2; //Same as (l+r)/2, but avoids overflow for
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);
        merge(arr, l, m, r);
    }
}

```

```

/* UTILITY FUNCTIONS */
/* Function to print an array */
void printArray(int A[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

/* Driver program to test above functions */
int main()
{
    int arr[] = {12, 11, 13, 5, 6, 7};
    int arr_size = sizeof(arr)/sizeof(arr[0]);

    printf("Given array is \n");
    printArray(arr, arr_size);

    mergeSort(arr, 0, arr_size - 1);

    printf("\nSorted array is \n");
    printArray(arr, arr_size);
    return 0;
}

```

Output:

```

Given array is
12 11 13 5 6 7

```

```

Sorted array is
5 6 7 11 12 13

```

Time Complexity: Sorting arrays on different machines. Merge Sort is a recursive algorithm and time complexity can be expressed as following recurrence relation.

Selection Sort · 7 hours ago

Sudhakar Mishra I think it should be $2n + 1$

Data Structures | Binary Trees | Question 12 · 8 hours ago

Sudhakar Mishra $(2n)!/(n+1)!n!$

Data Structures | Binary Trees | Question 6 · 1 day ago

Sudhakar Mishra Always Y will be more than one because after...

Data Structures | Stack | Question 7 · 1 day ago

AdChoices

► [Geeks Quiz](#)

► [Memory Quiz](#)

► [Quiz Application](#)

AdChoices

► [Quick Quiz](#)

► [C Programming Quiz Online](#)

► [Merge Data](#)

AdChoices

► [Convert Java Code to C++](#)

► [Java Array](#)

► [Memory Array](#)

$$T(n) = 2T(n/2) + \boxed{}$$

The above recurrence can be solved either using Recurrence Tree method or Master method.

It falls in case II of Master Method and solution of the recurrence is $\boxed{}$.

Time complexity of Merge Sort is $\boxed{}$ in all 3 cases (worst, average and best) as merge sort always divides the array in two halves and take linear time to merge two halves.

Auxiliary Space: $O(n)$

Algorithmic Paradigm: Divide and Conquer

Sorting In Place: No in a typical implementation

Stable: Yes

Applications of Merge Sort

1) Merge Sort is useful for sorting linked lists in $O(n \log n)$ time. Other $n \log n$ algorithms like Heap Sort, Quick Sort (average case $n \log n$) cannot be applied to linked lists.

2) Inversion Count Problem

3) Used in External Sorting

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Free C# Code Generator

 ironspeed.com

Create database & reporting apps
straight from your database! Try it



Related Questions:

- [Bubble Sort](#)
- [Selection Sort](#)
- [Binary Search](#)
- [QuickSort](#)
- [Heap Sort](#)
- [Insertion Sort](#)



7



0



1

13 Comments

GeeksQuiz

Sort by Best ▼



Join the discussion...



Archita · a year ago

Can you please give me the explanation of why in Merge sort we divide the l

Can you please give me the explanation of why in merge sort we divide the

5 ^ | v .



geeksquizgeeksquiz Mod → Archita · a year ago

If we divide in more parts, we may end up with more merge operati

3 ^ | v .



Ravi Prakash Giri · 3 months ago

What is the need of $j \leq n2$ for copying data into array $R[j]$? Can't it be done

1 ^ | v .



jimmy · 16 days ago

i think ,in the merge() function the second array

for($j = 0; j \leq n2; j++$)

$R[j] = arr[m + 1 + j];$

j should not be equal to $n2$.

what do you think?

^ | v .



GeeksforGeeks Mod → jimmy · 15 days ago

Thanks for pointing this out. We have updated the code.

^ | v .



amitav shaw · 23 days ago

it works with $j < n2$ also.. Can you explain why its $j \leq n2$ in the for loop for c unnecessarily creates an extra element in the array, but since the code does could be a possible bug or may be I'm wrong.

^ | v .



sameer · 2 months ago

can we create an array with some variable passed in it..!!

i mean not the constant

I mean not the constant.

like `int L[n1]..??`

0_o

^ | v .



Vin • 5 months ago

You have to free R and L in the end of merge function

^ | v .



GeeksforGeeks Mod → Vin • 4 months ago

Thanks for pointing this out. We have changed the code to use auto arrays.

^ | v .



Trish • a year ago

In the function "void mergesort" , we can define $m = l + (h-l)/2$ instead of $m = l$ indexes and that too of very high values, then we might have an overflow.

^ | v .



geeksquizgeeksquiz Mod → Trish • a year ago

Trish: Thanks for inputs. We have updated the post. Keep it up!

^ | v .



Mohamed Fasil → geeksquizgeeksquiz • 9 months ago

updatation has compilation error for h variable... it shd be r an

^ | v .



GeeksforGeeks Mod → Mohamed Fasil • 9 months ago

Thanks for pointing this out. We have changed h to r

^ | v .

 [Subscribe](#)

 [Add Disqus to your site](#)

Valid **XHTML Strict 1.0**

Powered by **WordPress & MooTools** | MiniMoo 1.3.4