



جامعة الأميرة نورة بنت عبد الرحمن
Princess Nourah bint Abdulrahman University



كلية علوم الحاسب والمعلومات
College of Computer and Information Sciences

Algorithms Design and Analysis, CS220T/D- PROJECT

Title: merge sort

Section: 4c5

The Students should
fill out these areas
"Yellow areas"

No.	Name	Academic number	Academic Email
	Monerah Almobarak	442002988	442002988@pnu.edu.sa
	Ahad Alotaibi	442004944	442004944@pnu.edu.sa
	Nada Alotaibi	442003374	442003374@pnu.edu.sa

Computer Sciences Department

First Semester 2019

Project

FOR INSTRUCTOR'S USE ONLY

Student name: **Monerah Almobarak**

Academic number: **442002988**

SO	Max. Marks	Marks Obtained	Note
1	4		
2	3		
6	3		
Total Score	10		

Student name: **Ahad Alotaibi**

Academic number: **442004944**

SO	Max. Marks	Marks Obtained	Note
1	4		
2	3		
6	3		
Total Score	10		

Student name: **Nada Alotaibi**

Academic number: **442003374**

SO	Max. Marks	Marks Obtained	Note
1	4		
2	3		
6	3		
Total Score	10		

Student name:

Academic number:

SO	Max. Marks	Marks Obtained	Note
1	4		
2	3		
6	3		
Total Score	10		

Introduction:

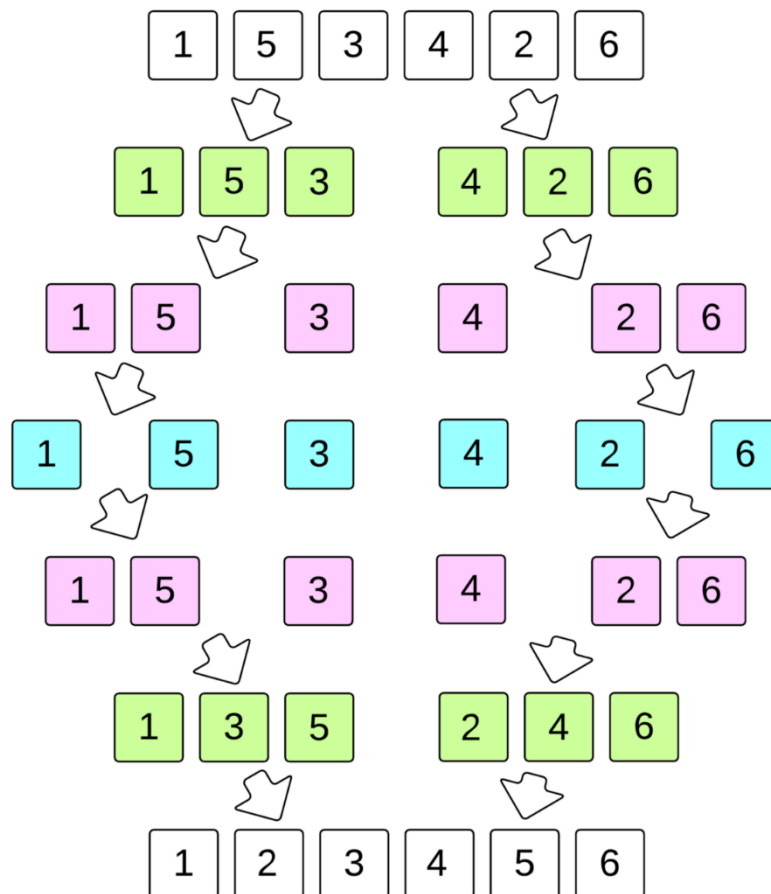
Merge sort is a divide-and-conquer algorithm based on the idea of breaking down a list into several sub-lists until each sub-list consists of a single element and merging those sub-lists in a manner that results into a sorted list.¹

Abstract: the Merge Sort algorithm closely follows the a divide and conquer paradigm. ²

Main idea: Merge Sort is a sorting algorithm, which is commonly used in computer science. Merge Sort is a divide and conquer algorithm. It works by recursively breaking down a problem into two or more sub-problems of the same or relate type, until these become simple enough to be solved directly.³

Merge Sort Algorithms Steps: ⁴

- 1- If it is only one element in the list it is already sorted, return.
- 2- Divide the list recursively into two halves until it can no more be divided.
- 3- Merge the smaller lists into new list in sorted order.



¹ GeeksforGeeks. 2022. Merge Sort for Linked Lists - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/merge-sort-for-linked-list/>> [Accessed 7 April 2022].

² : Cormen, T., Leiserson, C., Rivest, R. and Stein, C., n.d. Introduction to algorithms.

³ Bouchiha, A., 2022. Merge sort algorithm. [online] DEV Community. Available at: <<https://dev.to/ayabouchiha/merge-sort-algorithm-1i55>>

⁴ GeeksforGeeks. 2022. Merge Sort - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/merge-sort/>> [Accessed 25 March 2022].

Advantages of using merge sort algorithm:

- It is quicker for larger lists because unlike insertion and bubble sort it doesn't go through the whole list several times.
- It has a consistent running time, carries out different bits with similar times in a stage.⁵

Disadvantages of using merge sort algorithm:

- extra space to store subarrays
- slow for small arrays
- the algorithm does the whole process even the array is already sorted.⁶

analysis:

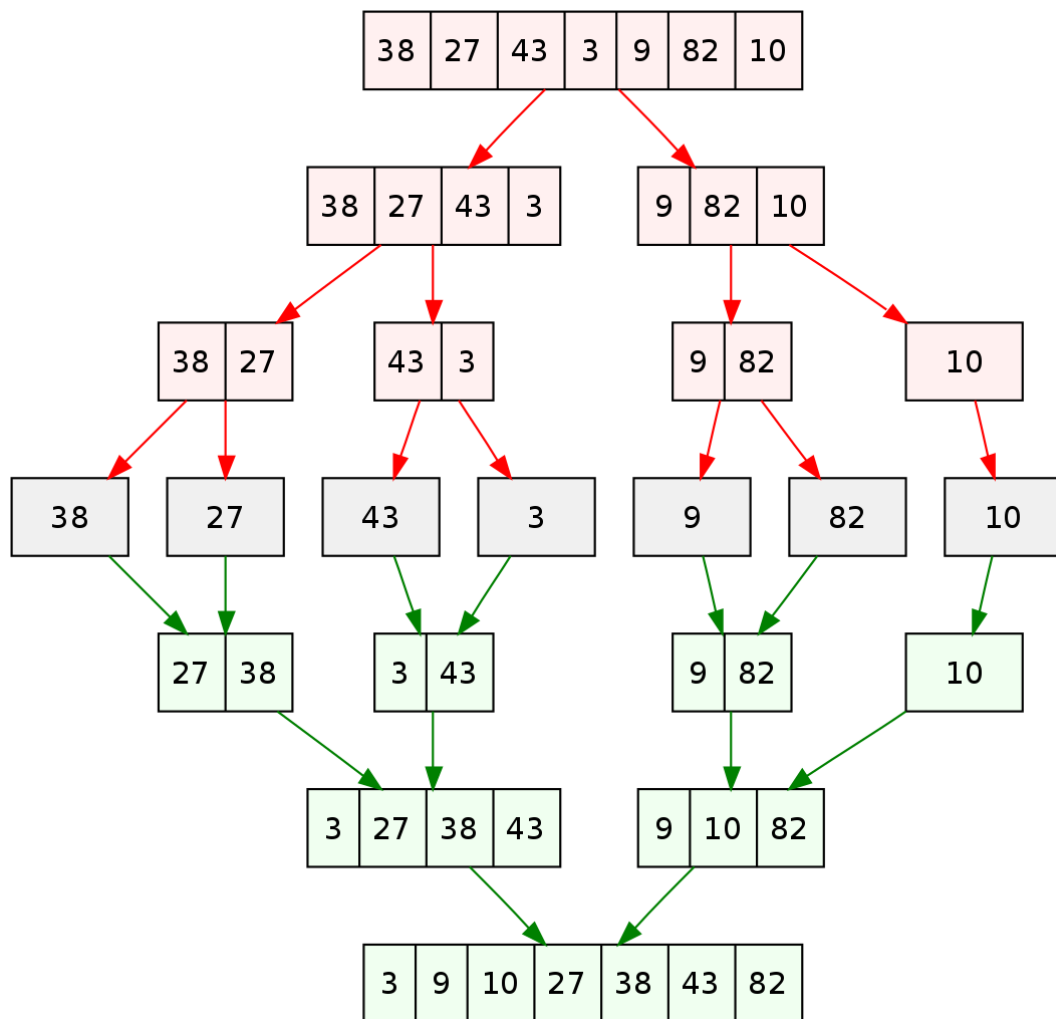
Merge Sort is a Divide and Conquer algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves. The merge() function is used for merging two halves. The merge(arr, l, m, r) is a key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted subarrays into one. MergeSort(arr[], l, m, r)

The following diagram from 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 into two halves till the size becomes 1. Once the size becomes 1, the merge processes come into action and start merging arrays back till the complete array is merged⁷

⁵ Getrevising.co.uk. 2022. Merge sort, advantages and disadvantages. [online] Available at: [Accessed 25 March 2022].

⁶ Medium. 2022. A Simplified Explanation of Merge Sort. [online] Available at: <<https://medium.com/karuna-sehgal/a-simplified-explanation-of-merge-sort-77089fe03bb2>>

⁷ GeeksforGeeks. 2022. Merge Sort for Linked Lists - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/merge-sort-for-linked-list/>> [Accessed 7 April 2022].



Running Time:⁸

- $T(n)$ **MERGE-SORT** $A[1 \dots n]$
- $\Theta(1)$ 1. If $n = 1$, done.
- $2T(n/2)$ 2. Recursively sort $A[1 \dots \lceil n/2 \rceil]$ and $A[\lceil n/2 \rceil + 1 \dots n]$.
- $\Theta(n)$ 3. **"Merge"** the 2 sorted lists

Recurrence for merge sort

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1; \\ 2T(n/2) + \Theta(n) & \text{if } n > 1. \end{cases}$$

⁸ : Cormen, T., Leiserson, C., Rivest, R. and Stein, C., n.d. Introduction to algorithms.

Kind of analysis:

Worst-case:

- $T(n)$ = maximum time of algorithm on any input of size n

Average-case:

- $T(n)$ = expected time of algorithm over all inputs of size n .

Best-case:

- $T(n)$ = minimum time of algorithm ((fastest time to complete, with optimal inputs chosen) ⁹

Worst-case performance	$O(n \log n)$
Best-case performance	$\Omega(n \log n)$ typical, $\Omega(n)$ natural variant
Average performance	$\Theta(n \log n)$

⁹ GeeksforGeeks. 2022. Merge Sort for Linked Lists - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/merge-sort-for-linked-list/>> [Accessed 7 April 2022].

Python Code Pseudocode for merge sort algorithm:¹⁰

```
def mergeSort(arr):
    if len(arr) > 1:

        # Finding the mid of the array
        mid = len(arr)//2

        # Dividing the array elements
        L = arr[:mid]

        # into 2 halves
        R = arr[mid:]

        # Sorting the first half
        mergeSort(L)

        # Sorting the second half
        mergeSort(R)

        i = j = k = 0

        # Copy data to temp arrays L[] and R[]
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1

        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1

        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1

    # Code to print the list

def printList(arr):
    for i in range(len(arr)):
        print(arr[i], end=" ")
    print()

# Driver Code
if __name__ == '__main__':
    arr = [12, 11, 13, 5, 6, 7]
    print("Given array is", end="\n")
    printList(arr)
    mergeSort(arr)
    print("Sorted array is: ", end="\n")
    printList(arr)
```

Output

```
Given array is
12 11 13 5 6 7
Sorted array is
5 6 7 11 12 13
```

¹⁰ GeeksforGeeks. 2022. Merge Sort - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/merge-sort/>> [Accessed 25 March 2022].

Pseudocode for merge sort algorithm¹¹

array first index mid index last index

```

MERGE(A, p, q, r)
1  n1 ← q − p + 1
2  n2 ← r − q
3  create arrays L[1 .. n1 + 1] and R[1 .. n2 + 1] → comment
4  for i ← 1 to n1
5      do L[i] ← A[p + i − 1] → left array
6  for j ← 1 to n2
7      do R[j] ← A[q + j] → right array
8  L[n1 + 1] ← ∞
9  R[n2 + 1] ← ∞ ] to fill the array ∞ at the end of the elements
10 i ← 1
11 j ← 1
12 for k ← p to r
13     do if L[i] ≤ R[j]
14         then A[k] ← L[i]
15             i ← i + 1
16         else A[k] ← R[j]
17             j ← j + 1

```

key

Our point of view:

Merge sort is one of the most efficient sorting algorithms in computer science, it's works on large numbers rather than small numbers, if we want to sort array has $n > 30$ elements its well be efficient for this elements.

¹¹ : Cormen, T., Leiserson, C., Rivest, R. and Stein, C., n.d. Introduction to algorithms.