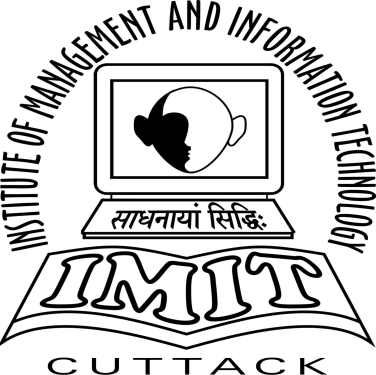
**PROJECT REPORT**

**On**

Submitted to Biju Patnaik University of Technology

in partial fulfillment of the requirements for the degree in

Master in Computer Application



**Submitted By**

**Haraprasad Moharana**

**Master in Computer Application**

**Regd. No. : 1705102046**

**INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY**

**Cuttack**

**DECLARATION**

I hereby declare that the application entitled “**How to rebound a object from a surface”** submitted to the Department of Computer application, **Institute of Management And Information Technology, Cuttack, Odisha** in partial fulfillment for the award of the degree of **Master in Computer Application** in session 2017-2020 is an authentic record of my own work with my project partners.

**Haraprasad Moharana**

(Reg. No.: 1705102040)

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report Of the MCA Minor Project Undertaken during MCA. This project in itself is an acknowledgement to the inspiration,drive and technical assistance contributed to it by many individuals.

Our heartiest thanks to Prof. Rayguru Akshay Kumar Das,HOD of MCA Department for providing us with an encouraging platform to develop this project.

We owe special debt of gratitude to Mr. Satyaprakash Swain, Mrs.Srutipragyan Swain and Mrs. Sujata Ray for their constant support and guidance through out the work of project.

Haraprasad Moharana

**CONTENTS**

1. Introduction
2. Divide And Conquer
3. Divide And Conquer Technique
4. Merge Sort Algorithm
5. Merge Sort Example
6. Hardware Specification
7. Software Specification
8. User Interface
9. Coding
10. Bibliography

**Title of the Project** : **Merge Sort**

**Introduction**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), merge sort is an efficient, general-purpose, [comparison-based](https://en.wikipedia.org/wiki/Comparison_sort) [sorting algorithm](https://en.wikipedia.org/wiki/Sorting_algorithm). Most implementations produce a [stable sort](https://en.wikipedia.org/wiki/Sorting_algorithm#Stability), which means that the order of equal elements is the same in the input and output. Merge sort is a [divide and conquer algorithm](https://en.wikipedia.org/wiki/Divide_and_conquer_algorithm) that was invented by [John von Neumann](https://en.wikipedia.org/wiki/John_von_Neumann) in 1945.

**Merge Sort:**

Merge Sort is a [Divide and Conquer](https://www.geeksforgeeks.org/divide-and-conquer-introduction/) 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.

**Divide And Conquer:**

* Merging two lists of one element each is the same as sorting them.
* Merge sort divides up an unsorted list until the above condition is met and then sorts the divided parts back together in pairs.
* Specifically this can be done by recursively dividing the unsorted list in half, merge sorting the left side then the right side and then merging the left and right back together.

Merge Sort Algorithm:

* Divide: Partition ‘n’ elements array into two sub lists with n/2 elements each
* Conquer: Sort sub list1 and sub list2.
* Combine: Merge sub list1 and sub list2.

**EXAMPLE:**

The following diagram shows the complete merge sort process for an example array {8, 2, 9, 4, 5, 3, 1,6}. If we take a closer look at the diagram, we can see that the array is recursively divided in two halfs 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.

## Hardware Specifications:

The implementation of the proposed system requires the following hardware specifications.

* 1. CPU Type : Any Pentium based systems.
  2. CPU Clock : 2.6 ghz or above
  3. RAM : 1 GB or above
  4. CD Drive : Any make
  5. Printer : U.P.S.[5 Kv ]: Any Make

## Software Specifications

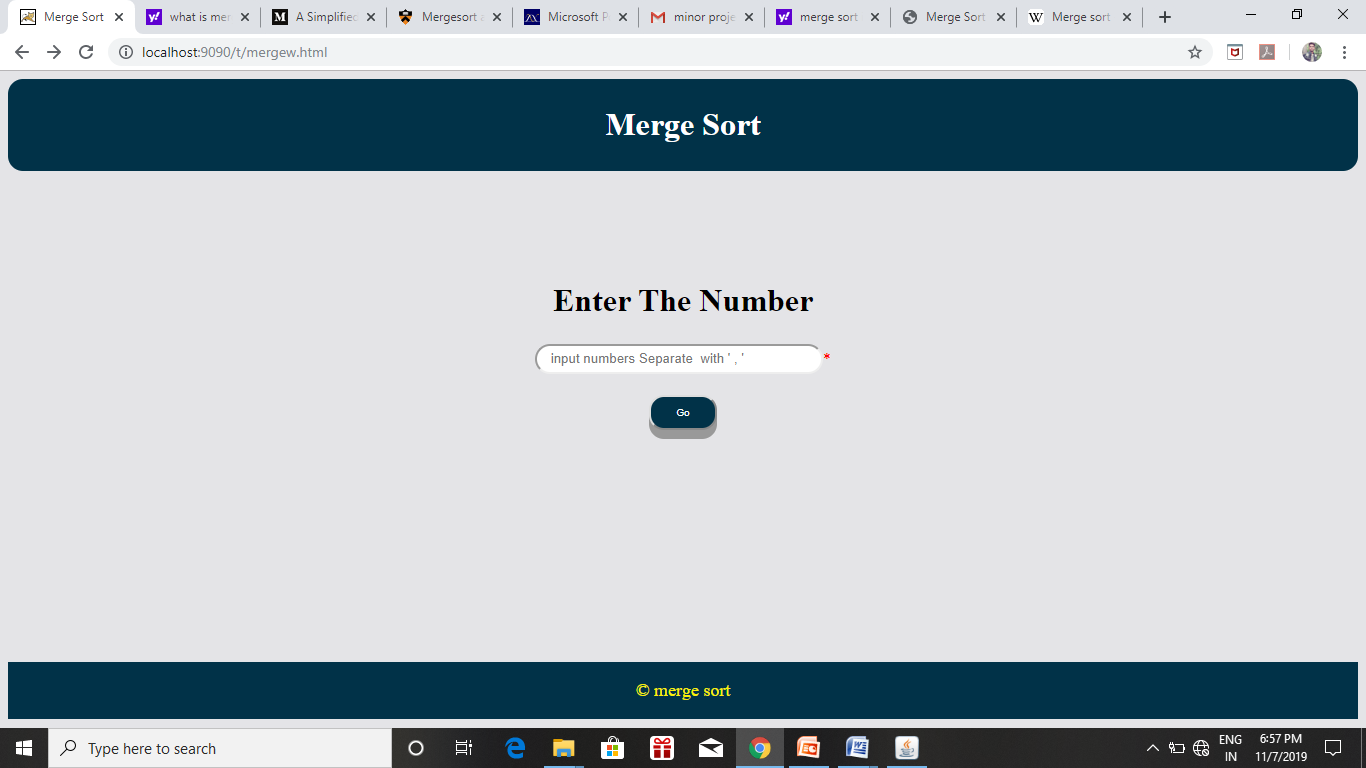
The software is designed under the following environment:

Language : Java EE

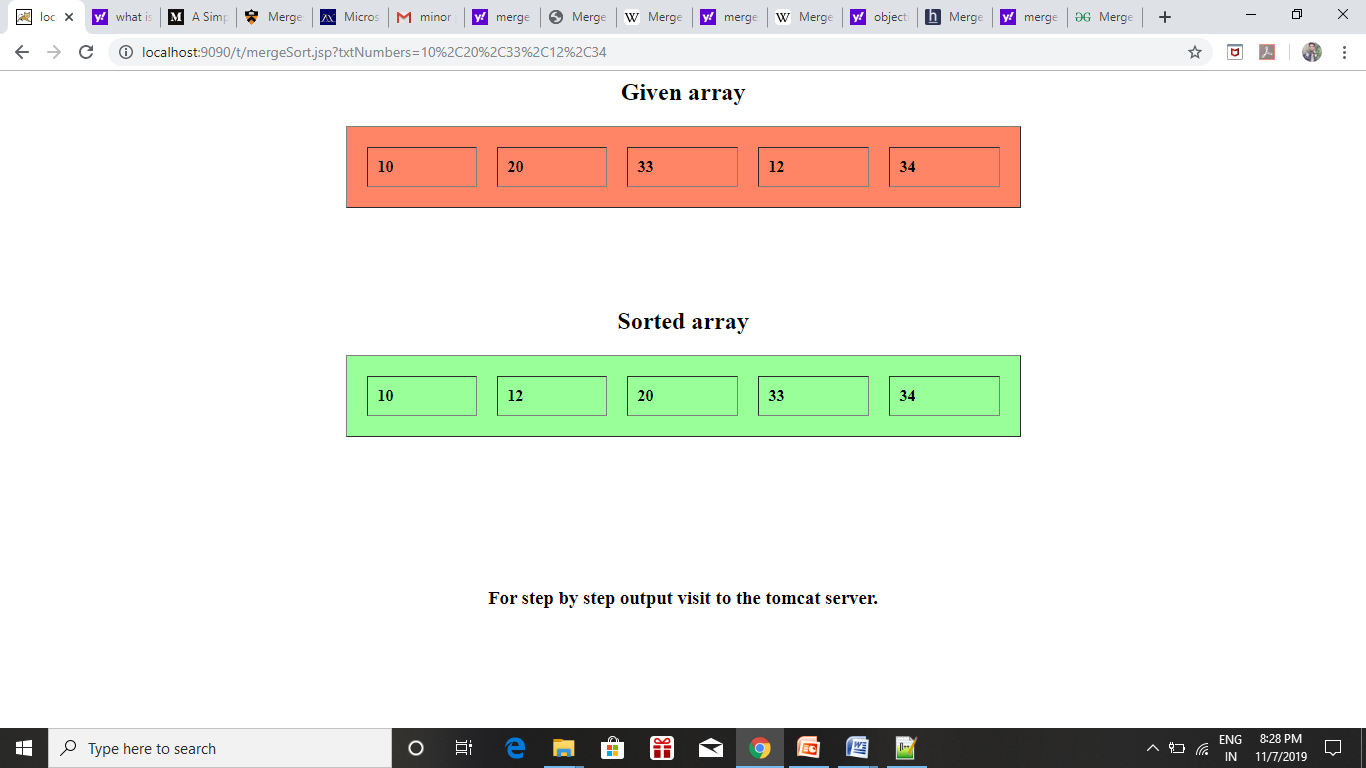
Operating System : Windows 10

Web Server : tomcat apache

**User Interface**

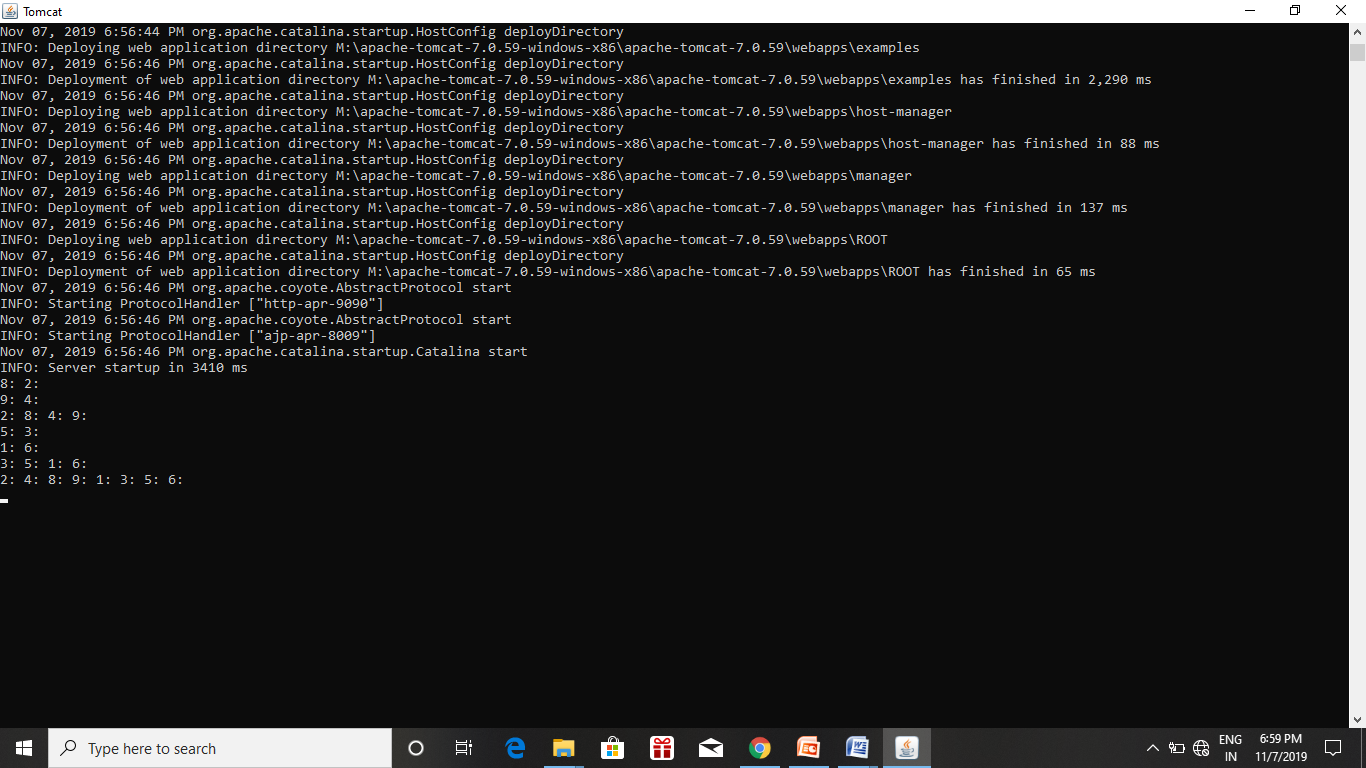
****

**Output**

****

**T**

**Tomcat Server Output**

****

**CODING:**

**Mergec.html:**

**<html>**

**<head>**

**<title>Merge Sort</title>**

**<style>**

**.header {**

**background-color:#013248;**

**padding: 6px;**

**text-align: center;**

**border-radius:15px;**

**}**

**.footer {**

**background-color: #013248;**

**text-align: center;**

**padding: 18px;**

**}**

**.button {**

**display: inline-block;**

**padding: 10px 25px;**

**font-size:10px;**

**cursor: pointer;**

**text-align: center;**

**color: #fff;**

**background-color: #013248;**

**border-radius: 15px;**

**box-shadow: 0 9px #999;**

**width:auto;**

**}**

**.button:hover {background-color: black}**

**.button:active {**

**background-color:black;**

**box-shadow: 0 2px #FBF004;**

**transform: translateY(3px);**

**}**

**input[type="text"],select{**

**border-radius:15px;**

**height:30px;**

**border:;**

**outline:none;**

**padding-left:10px;**

**}**

**</style>**

**</head>**

**<body bgcolor="#e4e4e7">**

**<div class="header"><h1><font color="white">Merge Sort</font></h1></div>**

**<form action="mergeSort.jsp">**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<center>**

**<h1><font>Enter The Number</font></h1>**

**<table>**

**<tr>**

**<td><input type="text" name="txtNumbers" size="35"placeholder=" input numbers Separate with ' , '" required><font color="red">\*</font></td>**

**</tr>**

**</table>**

**<br>**

**<div><input type="submit" value="Go" class="button"></div>**

**</center>**

**</form>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<br>**

**<div class="footer"><font face="Freestyle Script" color="#FBF004" size="4">&copy merge sort</font></div>**

**</body>**

**</html>**

**Merges.jsp**

**<%@page language="java" import="java.lang.\*" contentType="text/html"%>**

**<%!**

**int c,count,i,j;**

**String numbers[];**

**int num[];**

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

**{**

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

**int n2 = r - m;**

**int L[] = new int [n1];**

**int R[] = new int [n2];**

**for (int i=0; i<n1; ++i){**

**L[i] = arr[l + i];**

**//System.out.print(L[i]+", ");**

**}**

**// System.out.println();**

**for (int j=0; j<n2; ++j){**

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

**//System.out.print(R[j]+": ");**

**}**

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**for (int j=0; j<n1; ++j){**

**// R[j] = arr[m + 1+ j];**

**System.out.print(L[j]+": ");**

**}**

**for (int j=0; j<n2; ++j){**

**// R[j] = arr[m + 1+ j];**

**System.out.print(R[j]+": ");**

**}**

**System.out.println();**

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**int i = 0, j = 0;**

**int k = l;**

**while (i < n1 && j < n2)**

**{**

**if (L[i] <= R[j])**

**{**

**arr[k] = L[i];**

**i++;**

**//System.out.println("h"+arr[k]);**

**}**

**else**

**{**

**arr[k] = R[j];**

**j++;**

**//System.out.println("t"+arr[k]);**

**}**

**k++;**

**}**

**while (i < n1)**

**{**

**arr[k] = L[i];**

**i++;**

**k++;**

**}**

**while (j < n2)**

**{**

**arr[k] = R[j];**

**j++;**

**k++;**

**}**

**}**

**public void sort(int arr[], int l, int r)**

**{**

**if (l < r)**

**{**

**int m = (l+r)/2;**

**sort(arr, l, m);**

**sort(arr , m+1, r);**

**merge(arr, l, m, r);**

**}**

**}**

**static void printArray(int arr[])**

**{**

**int n = arr.length;**

**for (int i=0; i<n; ++i)**

**System.out.print(arr[i] + " ");**

**System.out.println();**

**}**

**%>**

**<%**

**numbers =new String[20];**

**numbers =request.getParameter("txtNumbers").split(",");**

**//given array**

**out.println("<h2 align='center'>Given array</h2>");**

**out.println("<table width='50%' border='1px' align='center' cellspacing='20px' cellpadding='10px' style='background:#ff8566;'>");**

**out.println("<tr>");**

**for(i=0;i<numbers.length;i++){**

**out.println("<td><b>"+numbers[i]+"</b></td>");**

**}**

**out.println("</tr>");**

**out.println("</table>");**

**num = new int[numbers.length];**

**for(int c=0;c<numbers.length;c++)**

**{**

**num[c] = Integer.parseInt(numbers[c]);**

**}**

**sort(num,0,num.length-1);**

**out.println("<h2 align='center' style='margin-top:100px;'>Sorted array</h2>");**

**out.println("<table width='50%' border='1px' align='center' cellspacing='20px' cellpadding='10px' style='background:#99ff99;'>");**

**out.println("<tr>");**

**for(int i=0;i<num.length;i=i+1){**

**out.println("<td width='10%'><b>"+num[i]+"</b></td>");**

**}**

**out.println("</tr>");**

**out.println("</table>");**

**out.println("<h3 align='center' style='margin-top:150px;'>For step by step output visit to the tomcat server.</h3>");**

**// printArray(num);**

**%>Bibliography**

The following books have been referred during the preparation of this project.

1. A Complete guide to HTML &CSS
2. Java Server Pages