MERGE SORT

Nikhil R S USN: 1BM15IS051
K. Sai Supreeth USN:1BM15IS034

Topics to be covered:

- Introduction
- Definition
- Algorithm
- Steps involved
- Program
- Applications

Introduction:

- Merge Sort is a complex and fast sorting algorithm that repeatedly divides an un-sorted section into two equal sub-sections, sorts them separately and merges them correctly.

Definition:

- Merge sort 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.

Steps involved:

- ***Divide*** the problem into sub-problems that are similar to the original but smaller in size.
- ***Conquer*** the sub-problems by solving them recursively. If they are small enough, just solve them in a straightforward manner.
- ***Combine*** the solutions to create a solution to the original problem.

Algorithm:

mergeSort(arr[], l, r)

If l < r

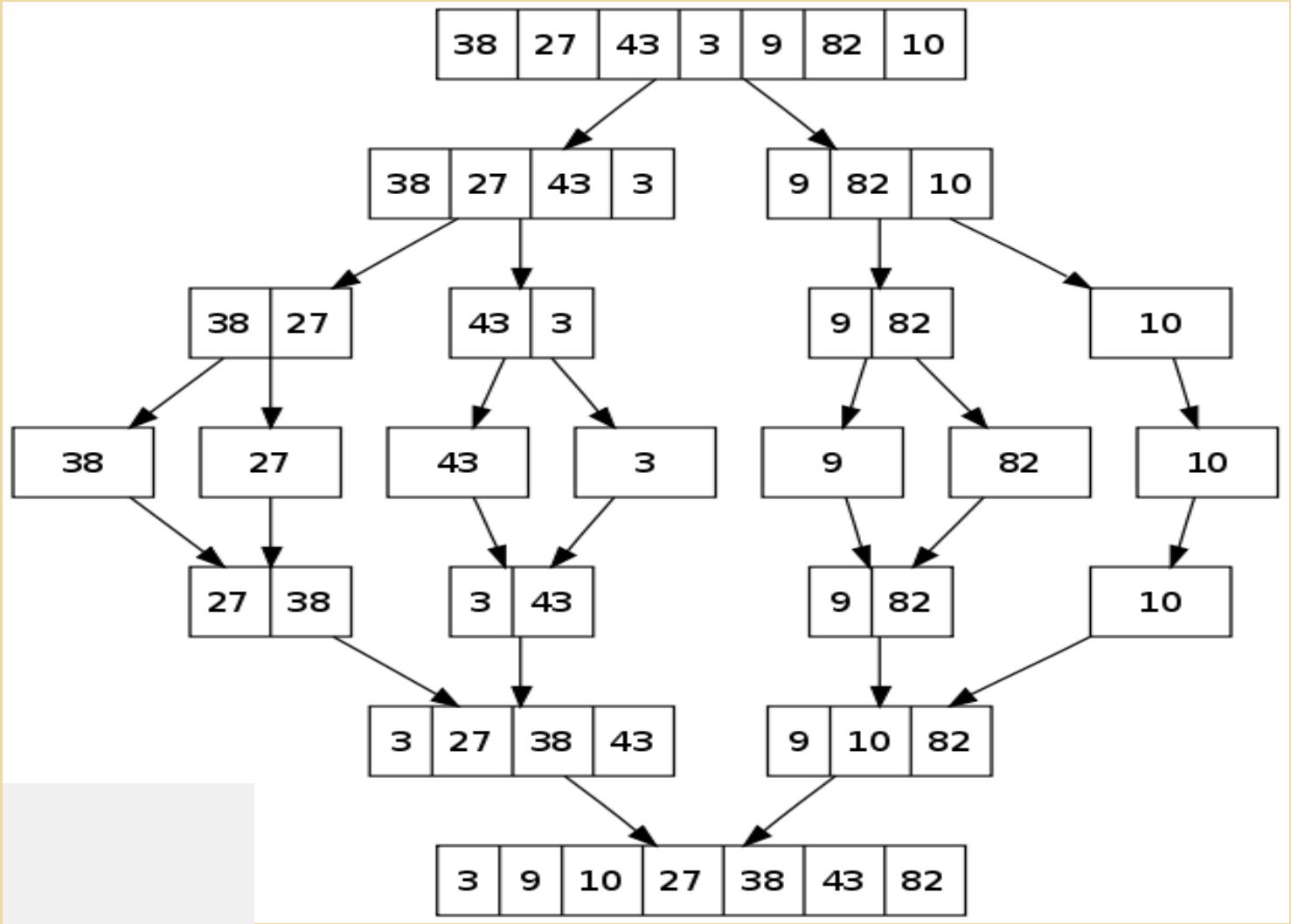
1. Find the middle point to divide the array into two halves: middle m = (l+r)/2

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)

Example:


38 27 43 3 9 82 10
38 27 43 3
9 82 10
38 27
43 3
9 82
10
38
27
43
3
9
82
10
27 38
3 43
9 82
10
3 27 38 43
9 10 82
3 9 10 27 38 43 82

Program:

```
#include <iostream>
using namespace std;
void merge(int a[], int low, int high, int mid)
{
    int i, j, k, c[50];
    i = low;
    k = low;
    j = mid + 1;
    while (i <= mid && j <= high)
        //Compares the elements of two subarrays and merges then
    {
        if (a[i] < a[j])
        {
            c[k] = a[i];
            k++;
            i++;
        }
        else
        {
            c[k] = a[j];
            k++;
            j++;
        }
    }
```

```
    while (i <= mid)    //Copies the remaining elements of
                        //left array,if there is any
    {
        c[k] = a[i];
        k++;
        i++;
    }
    while (j <= high)   //Copies the remaining elements of
                        //right array,if there is any
    {
        c[k] = a[j];
        k++;
        j++;
    }
    for (i = low; i < k; i++)
    {
        a[i] = c[i];
    }
}
```

```
void mergesort(int a[], int low, int high)
{
     int mid;
     if (low < high)
     {
          mid=(low+high)/2;
          mergesort(a,low,mid);    //Recursion call till low<high
          mergesort(a,mid+1,high);
          merge(a,low,high,mid);
     }
     return;
}
```

```
int main()
{
    int a[20], i;
    cout<<"Enter  the elements\n";
    for (i = 0; i < 5; i++)
    {
        cin>>a[i];
    }
    mergesort(a, 0, 4);
    cout<<"Sorted array\n";
    for (i = 0; i < 5; i++)
    {
        cout<<a[i];
        cout<<"\n";
    }
    return 0;
}
```

Output:

```
Enter  the elements
8
6
11
12
14
Sorted array
6
8
11
12
14
Program ended with exit code: 0
```

Why Merge Sort??

- Compared to insertion sort merge sort is faster.
- On small inputs, insertion sort may be faster. But for large enough inputs, merge sort will always be faster, because its running time grows more slowly than insertion sorts.

Applications:

- Merge sort type algorithms allows large data sets to be sorted easily.
- Merge sort accesses data sequentially and the need of random access is low.
- Inversion count problem.
- Used in External Sorting.

- Organize an MP3 library.

	Name	Artist	Time	Album
12	Let It Be	The Beatles	4:03	Let It Be
13	Take My Breath Away	BERLIN	4:13	Top Gun - Soundtrack
14	Circle Of Friends	Better Than Ezra	3:27	Empire Records
15	Dancing With Myself	Billy Idol	4:43	Don't Stop
16	Rebel Yell	Billy Idol	4:49	Rebel Yell
17	Piano Man	Billy Joel	5:36	Greatest Hits Vol. 1
18	Pressure	Billy Joel	3:16	Greatest Hits, Vol. II (1978 - 1985) (Disc 2)
19	The Longest Time	Billy Joel	3:36	Greatest Hits, Vol. II (1978 - 1985) (Disc 2)
20	Atomic	Blondie	3:50	Atomic: The Very Best Of Blondie
21	Sunday Girl	Blondie	3:15	Atomic: The Very Best Of Blondie
22	Call Me	Blondie	3:33	Atomic: The Very Best Of Blondie
23	Dreaming	Blondie	3:06	Atomic: The Very Best Of Blondie
24	Hurricane	Bob Dylan	8:32	Desire
25	The Times They Are A-Changin'	Bob Dylan	3:17	Greatest Hits
26	Livin' On A Prayer	Bon Jovi	4:11	Cross Road
27	Beds Of Roses	Bon Jovi	6:35	Cross Road
28	Runaway	Bon Jovi	3:53	Cross Road
29	Rasputin (Extended Mix)	Boney M	5:50	Greatest Hits
30	Have You Ever Seen The Rain	Bonnie Tyler	4:10	Faster Than The Speed Of Night
31	Total Eclipse Of The Heart	Bonnie Tyler	7:02	Faster Than The Speed Of Night
32	Straight From The Heart	Bonnie Tyler	3:41	Faster Than The Speed Of Night
33	Holding Out For A Hero	Bonny Tyler	5:49	Meat Loaf And Friends
34	Dancing In The Dark	Bruce Springsteen	4:05	Born In The U.S.A.
35	Thunder Road	Bruce Springsteen	4:51	Born To Run
36	Born To Run	Bruce Springsteen	4:30	Born To Run
37	Jungleland	Bruce Springsteen	9:34	Born To Run

- Display Google PageRank results.
- The e-commerce application.

ThankYou…!!!!