

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**DIVIDE AND CONQUER**

**A MINOR PROJECT REPORT**

***Submitted by***

**SEETHAMSETTI SRIKANTH- RA2011042010088**

**BIJJAM VISWANATH REDDY – RA2011042010092**

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND BUSINESS SYSTEMS**

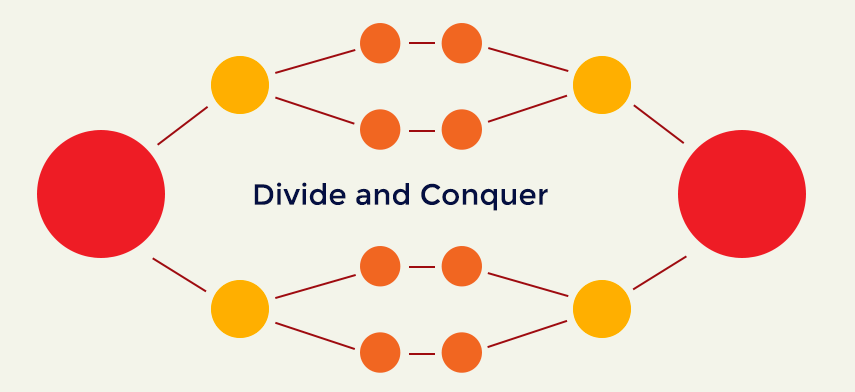
**Faculty Name: Prabhu Kavin B**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS**

DAA PROJECT

**DIVIDE AND CONQUER**



**Contents**

[PROBLEM DEFINITION:](#_TOC_250003)

[PROBLEM EXPLANATION:](#_TOC_250002)

DESIGN TECHNIQUE:

GENERAL TECHNIQUE: …………………..

ALGORITHM: .......................

ALGORITHM EXPLANATION WITH EXAMPLE: .......................

CODE: ………………..

[COMPLEXITY ANALYSIS](#_TOC_250001)

[CONCLUSION](#_TOC_250000)

# PROBLEM DEFINITION:

* Due to the updating of technology in day to day life there are employees in the company who have out-dated skills. To update skills of the employees the company has planned to send their employees to the training session.
* These were reviewed by advanced team. Divide the selected employees into two groups one having less skill compared to the other

**INPUT FORMAT:**

* The first line contains ‘n’ it means the no of employees for training and reviews of the employees denoted as re[i]

**OUTPUT FORMAT:**

* Print the two groups of people having less skills compared to other

# 

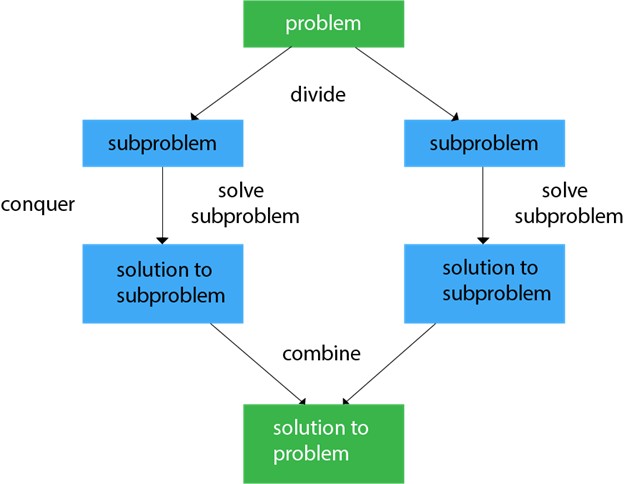
# PROBLEM EXPLANATION WITH DIAGRAM AND EXAMPLE:

* THE QUESTION IS ABOUT THERE ARE SOME EMPLOYEES WHO HAVE OUTDATED SKILLS SO THE MANAGER OF THE COMPANY WANTS TO KNOW HIS EMPLOYEES WHO HAVE OUTDATED SKILLS.
* THE MANAGER APPOINTS AN ADVANCED TEAM TO REVIEW
* THE EMPLOYEES IN A SCALE OF 1 TO 10. THEN HE CONDUCTED A TRAINING SESSION.
* HERE THE MANAGER WILL DIVIDE THE EMPLOYEES INTO TWO EQUAL GROUPS ONE GROUP WHICH CONTAINS EMPLOYEES WITH LESSER REVIEWS AND OTHER GROUP WITH BIT HIGHER REVIEWS.
* SO, THE GROUP1 WILL GO TO THE HIGHER LEVEL OF TRAINING SESSION AND OTHER WILL GO TO MEDIUM LEVEL OF TRAINING SESSION.

# DESIGN TECHNIQUES USED:

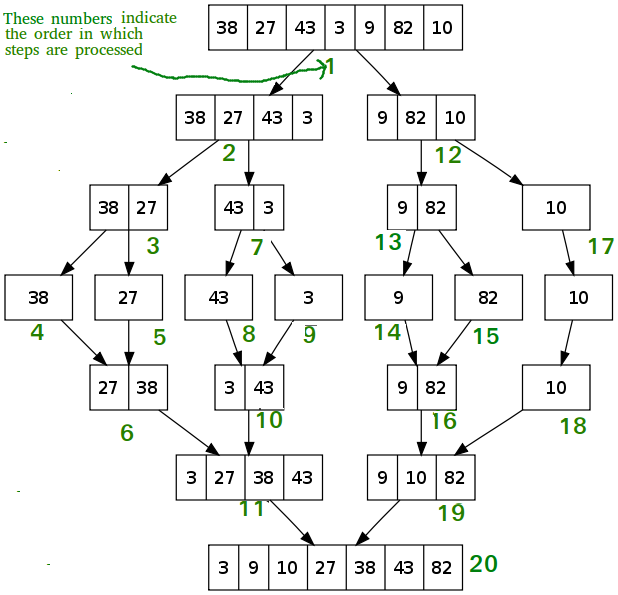
## DIVIDE AND CONQUER:

Divide and Conquer is an algorithmic pattern. In algorithmic methods, the design is to take a dispute on a huge input, break the input into minor pieces, decide the problem on each of the small pieces, and then merge the piecewise solutions into a global solution. This mechanism of solving the problem is called the Divide & Conquer Strategy.



**GENARAL TECHNIQUE:**

A classic example of Divide and Conquer is Merge Sort demonstrated below. In Merge Sort, we divide array into two halves, sort the two halves recursively, and then merge the sorted halves.





**MERGE SORT:**

It divides the input array into two halves, calls itself for the two halves, and then it 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 sub-arrays into one.

# ALGORITHM:

Divide and Conquer is used in this problem.

## Approach:

To find the maximum and minimum element from a given array is an application for divide and conquer. In this problem, we will find the maximum and minimum elements in a given array. In this problem, we are using a divide and conquer approach(DAC) which has three steps divide, conquer and combine..

## Algorithm:

A divide and conquer algorithm is a strategy of solving a large problem by

1. breaking the problem into smaller sub-problems
2. solving the sub-problems, and
3. combining them to get the desired output.

To use the divide and conquer algorithm, recursion is used.

Divide and Conquer algorithm consists of a dispute using the following three steps.

**Divide** the original problem into a set of sub problems.

**Conquer:** Solve every sub problem individually, recursively.

**Combine:** Put together the solutions of the sub problems to get the solution to the whole problem.

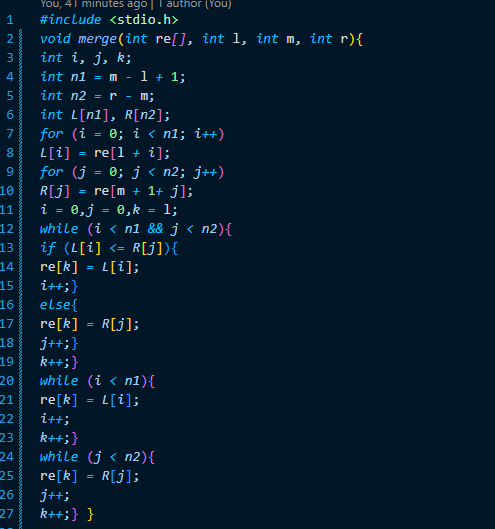
# ALGORITHM EXPLANATION WITH EXAMPLE:

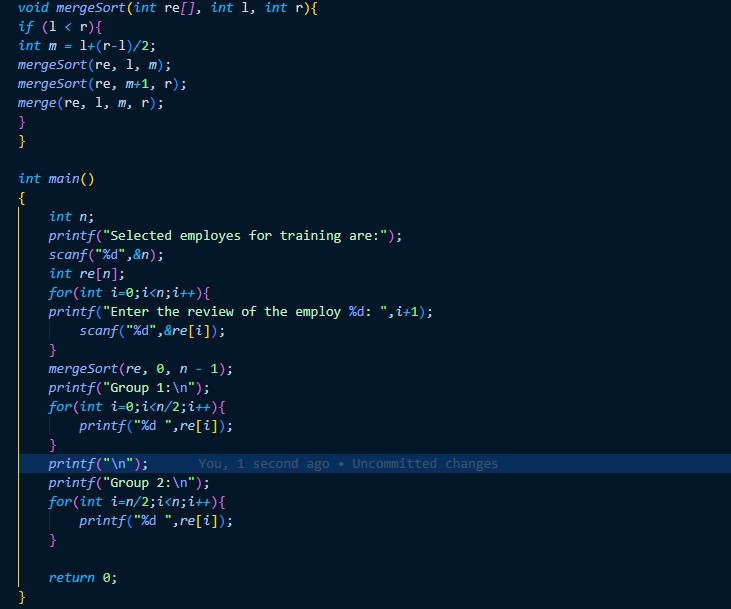
* First the algorithm divides the array into two parts.

For eg: 8,7,1,2,4,3,9,4,7,2 divide into 8,7,1,2,4 and 3,9,4,7,2

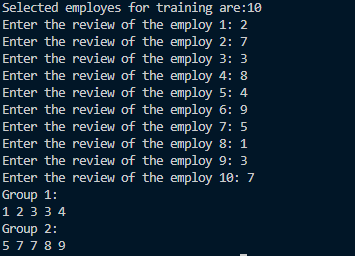
* This goes on dividing until l is grater then right.
* Now create two empty arrays and inset the numbers of the left side of the array into one array and another side of elements into another array.
* Now merge the array into one after sorting.

**CODE:-**

****

****

**OUTPUT:-**



# COMPLEXITY ANALYSIS:

The complexity of the divide and conquer algorithm is calculated using the master theorem.

T(n) = aT(n/b) + f(n),

where,

n = size of input

a = number of subproblems in the recursion

n/b = size of each subproblem. All subproblems are assumed to have the same size.

f(n) = cost of the work done outside the recursive call, which includes the cost of dividing the problem and cost of merging the solutions

T(n) = aT(n/b) + f(n)

= 2T(n/2) + O(n)

Where,

a = 2 (each time, a problem is divided into 2 subproblems)

n/b = n/2 (size of each sub problem is half of the input)

f(n) = time taken to divide the problem and merging the subproblems

T(n/2) = O(n log n) (To understand this, please refer to the master theorem.)

Now, T(n) = 2T(n log n) + O(n)

≈ O(n log n)

# CONCLUSION:

We have created a solution for a real-life scenario completely from scratch. It helps us to write logic and maintain clean structure in code. This helps in creating solutions for real time problems in our day-to-day life. The problem was solved efficiently using divide and conquer technique.