S. B. JAIN INSTITUTE OF TECHNOLOGY,

MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 1

Aim: Virtual Lab: Apply the knowledge of divide and conquer strategy and how it can be used to solve the sorting problem using merge sort.

Name of Student: Shrutika Pradeep Bagdi

Roll No: CS22130

Semester/Year: 5th /3rd

Academic Session: 2024-2025

Date of Performance:

Date of Submission:

AIM: Virtual Lab: Apply the knowledge of divide and conquer strategy and how it can be used to solve the sorting problem using merge sort. (https://ds1-iiith.vlabs.ac.in/exp/merge-sort/index.html)

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- To successfully understand the concept of divide and conquer.
- To demonstrate the working of merge sort.
- To analyze the time complexity of merge sort.

THEORY:

Divide and Conquer Algorithm is a problem-solving technique used to solve problems by dividing the main problem into subproblems, solving them individually and then merging them to find solution to the original problem.

Divide and Conquer Algorithm involves breaking a larger problem into smaller subproblems, solving them independently, and then combining their solutions to solve the original problem. The basic idea is to recursively divide the problem into smaller subproblems until they become simple enough to be solved directly. Once the solutions to the subproblems are obtained, they are then combined to produce the overall solution.

1. Divide:

- Break down the original problem into smaller subproblems.
- Each subproblem should represent a part of the overall problem.
- The goal is to divide the problem until no further division is possible.

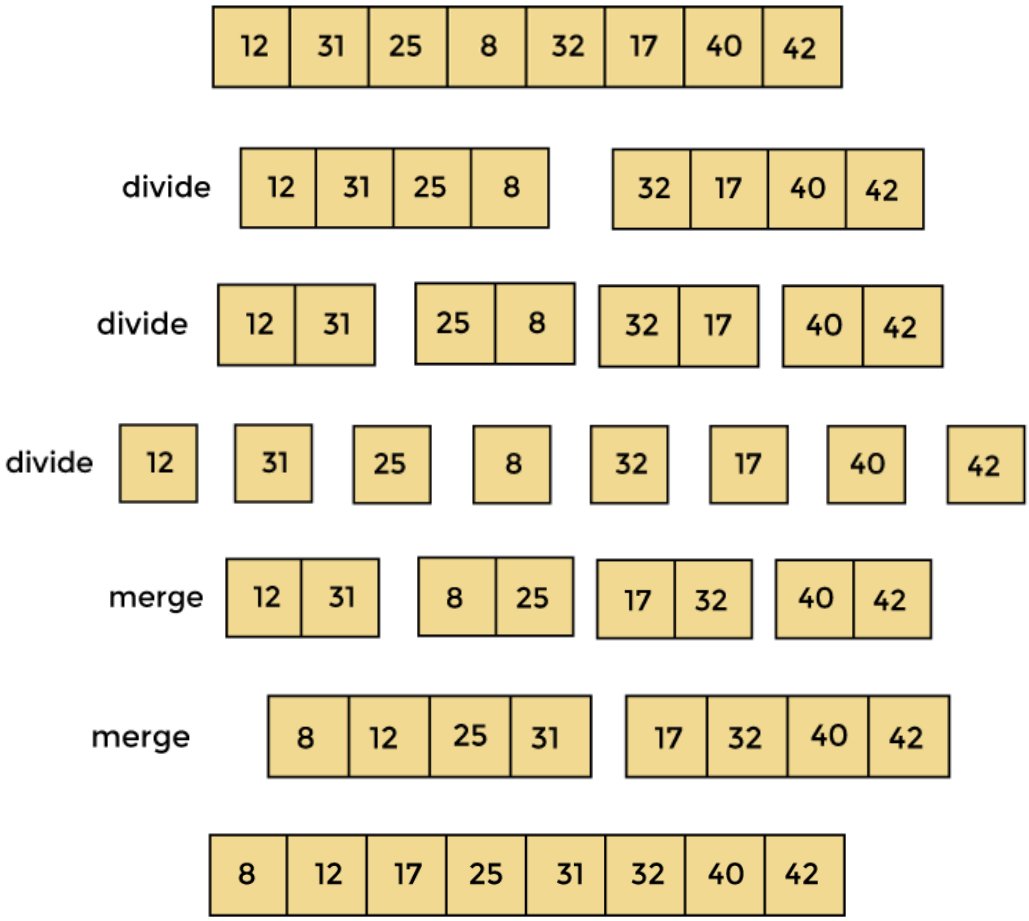
2. Conquer:

- Solve each of the smaller subproblems individually.
- If a subproblem is small enough (often referred to as the "base case"), we solve it directly without further recursion.
- The goal is to find solutions for these subproblems independently.

3. Merge:

- Combine the sub-problems to get the final solution of the whole problem.
- Once the smaller subproblems are solved, we recursively combine their solutions to get the solution of larger problem.
- The goal is to formulate a solution for the original problem by merging the results from the subproblems.

Example of Merge Sort:



ALGORITHM:

MERGE_SORT(arr, beg, end)

if *beg < end*
set mid = (beg + end)/2
MERGE_SORT(arr, beg, mid)
MERGE_SORT(arr, mid + 1, end)
MERGE (arr, beg, mid, end)
end of ***if***

END MERGE_SORT

SCREENSHOTS OF DEMONSTRATION:

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Computer Science and Engineering > Data Structures – 1 > Experiments

- Aim
- Overview
- Recap
- Pretest
- Divide and Conquer
- Merge Sort Algorithm
- Analysis
- Posttest
- Further Readings/References
- Feedback

Merge Sort

Choose difficulty: ☑ Beginner ☑ Intermediate

1. Which of the following is not an array?
   - a: [1, 4, 3, 10] Explanation
   - b: ["A", True, "Yes", "Hello world"] Explanation
   - c: ["Test", "False", "x", "hi"] Explanation
   - d: [True, False, False, True] Explanation

2. Which of the following is an array sorted in descending order?
   - a: -17, -14, -8, -19 Explanation
   - b: -17, 20, 20, 100 Explanation
   - c: 900, 14, -100, -1 Explanation
   - d: 100, 100, 10, 0 Explanation

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

d: 100, 100, 10, 0 Explanation

3. Consider the following arrays:
   A = [5, -5, 10, 1]
   B = [0, 100, 1, 0]
   Which of the following arrays represents the combination of the 2 arrays in a sorted order(assume ascending order)?
   - a: [5, -5, 10, 1, 0, 100, 1, 0] Explanation
   - b: [0, 100, 1, 0, 5, -5, 10, 1] Explanation
   - c: [0, 0, 1, 1, -5, 5, 10, 100] Explanation
   - d: [-5, 0, 0, 1, 1, 5, 10 ,100] Explanation

4. When dividing an array into equal sized partitions (as closely as possible), how does the time complexity of the overall algorithm vary with the number of partitions made per step (fixed number per step)? Note that the number of partitions directly affects the size of the partitions. Greater the number of partitions, the smaller each partition will be.
   - a: Does not vary, i.e, is independent of the number of partitions made per step Explanation
   - b: Depends on the situation Explanation
   - c: Is inversely proportional, i.e, the time complexity decreases with increase in the number of partitions being made per step Explanation
   - d: Is directly proportional, i.e, the time complexity increases with increase in the number of partitions being made per step Explanation

Submit Quiz

4 out of 4

Virtual Labs
An MoE Govt of India Initiative

Merge Sort

Instructions

Setup Reset Next

Observations

Done sorting!

2	5	10	12	17	22
0	1	2	3	4	5

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Overview
Recap
Pretest
Divide and Conquer
Merge Sort Algorithm
Aim
Concept
Demo
Practice
Exercise
Quiz
Analysis
Posttest
Further Readings/References
Feedback

Merge Sort

1. Consider the following array:
A = [9, 100, -1, 4, 8, 6, 9]
Determine the number of iterations of division (same as the number of iterations of merging) as well as the total number of division operations that will be required for the above array.
- a: 2 and 6 Explanation
- b: 3 and 6 Explanation
- c: 6 and 2 Explanation
- d: 6 and 3 Explanation

2. During which step does the actual sorting occur?
- a: Division Explanation
- b: Separate step for sorting Explanation
- c: Merging Explanation
- d: None of the above Explanation

3. Which of the following accurately describes the merge step in the algorithm (assume ascending order sort)?
- a: We directly merge with 1 sub-array after the other. Explanation
- b: We iterate through each of the concerned sub-arrays simultaneously where we add the smaller of the elements under consideration at that point to the merged array and advance the index for the array from which the element was selected. Explanation
- c: We alternate between the elements of the two sub-arrays, i.e. first we pick one element from the 1st sub-array and then pick another element at the same index in the other sub-array. Explanation
- d: None of the above. Explanation

Submit Quiz
3 out of 3

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

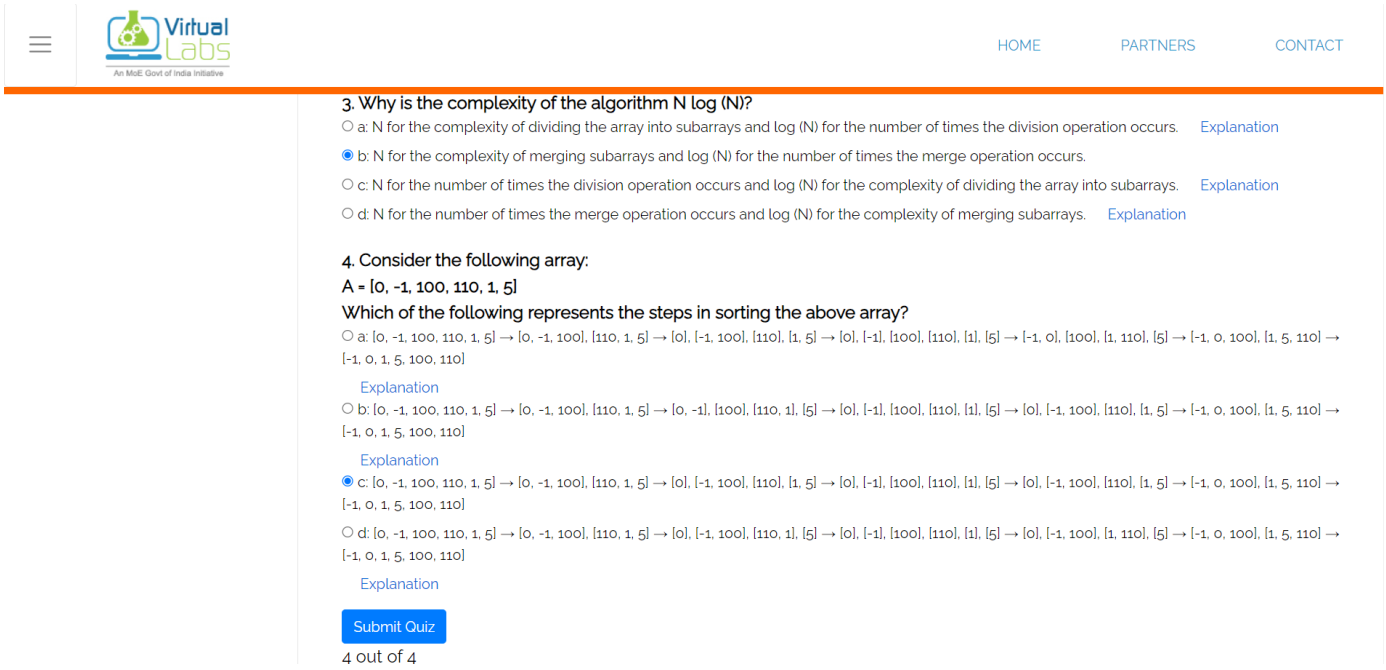
Aim
Overview
Recap
Pretest
Divide and Conquer
Merge Sort Algorithm
Analysis
Posttest
Further Readings/References
Feedback

Merge Sort

Choose difficulty: ☑ Beginner ☑ Intermediate ☑ Advanced

1. What will be the space complexity of the algorithm?
- a: O(N) Explanation
- b: O(2^N) Explanation
- c: O(1) Explanation
- d: O(N^2) Explanation

2. Consider the following array:
A = [8, 7, -2, 4, 1, 100, 0, -1]
Which of the following subarray pairs are merged with each other during one of the merge operations (assume ascending order sort)?
- a: [-2, 4] and [1, 100] Explanation
- b: [7, 8] and [-2, 4]
- c: [8, 7] and [-2, 4] Explanation
- d: [1, 4] and [0, 100] Explanation



Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

3. Why is the complexity of the algorithm N log (N)?

- a: N for the complexity of dividing the array into subarrays and log (N) for the number of times the division operation occurs. Explanation
- b: N for the complexity of merging subarrays and log (N) for the number of times the merge operation occurs.
- c: N for the number of times the division operation occurs and log (N) for the complexity of dividing the array into subarrays. Explanation
- d: N for the number of times the merge operation occurs and log (N) for the complexity of merging subarrays. Explanation

4. Consider the following array:
A = [0, -1, 100, 110, 1, 5]
Which of the following represents the steps in sorting the above array?

- a: [0, -1, 100, 110, 1, 5] → [0, -1, 100], [110, 1, 5] → [0], [-1, 100], [110], [1, 5] → [0], [-1], [100], [110], [1], [5] → [-1, 0], [100], [1, 110], [5] → [-1, 0, 100], [1, 5, 110] → [-1, 0, 1, 5, 100, 110]
  Explanation
- b: [0, -1, 100, 110, 1, 5] → [0, -1, 100], [110, 1, 5] → [0, -1], [100], [110, 1], [5] → [0], [-1], [100], [110], [1], [5] → [0], [-1, 100], [110], [1, 5] → [-1, 0, 100], [1, 5, 110] → [-1, 0, 1, 5, 100, 110]
  Explanation
- c: [0, -1, 100, 110, 1, 5] → [0, -1, 100], [110, 1, 5] → [0], [-1, 100], [110], [1, 5] → [0], [-1], [100], [110], [1], [5] → [0], [-1, 100], [110], [1, 5] → [-1, 0, 100], [1, 5, 110] → [-1, 0, 1, 5, 100, 110]
- d: [0, -1, 100, 110, 1, 5] → [0, -1, 100], [110, 1, 5] → [0], [-1, 100], [110, 1], [5] → [0], [-1], [100], [110], [1], [5] → [0], [-1, 100], [1, 110], [5] → [-1, 0, 100], [1, 5, 110] → [-1, 0, 1, 5, 100, 110]
  Explanation

Submit Quiz

4 out of 4

CONCLUSION:

Here, successfully gained knowledge of the divide and conquer strategy and used it to solve the sorting problem with merge sort.

DISCUSSION AND VIVA VOCE:

- Explain the divide and conquer strategy.
- Explain the working of merge sort.
- Discuss the time complexity and space complexity of merge sort.

REFERENCES:

- *https://ds1-iiith.vlabs.ac.in/exp/merge-sort/index.html*
- *https://www.geeksforgeeks.org/introduction-to-divide-and-conquer-algorithm/#divide-and-conquer-algorithm-definition*
- *https://www.javatpoint.com/merge-sort*
- *https://www.geeksforgeeks.org/merge-sort/*