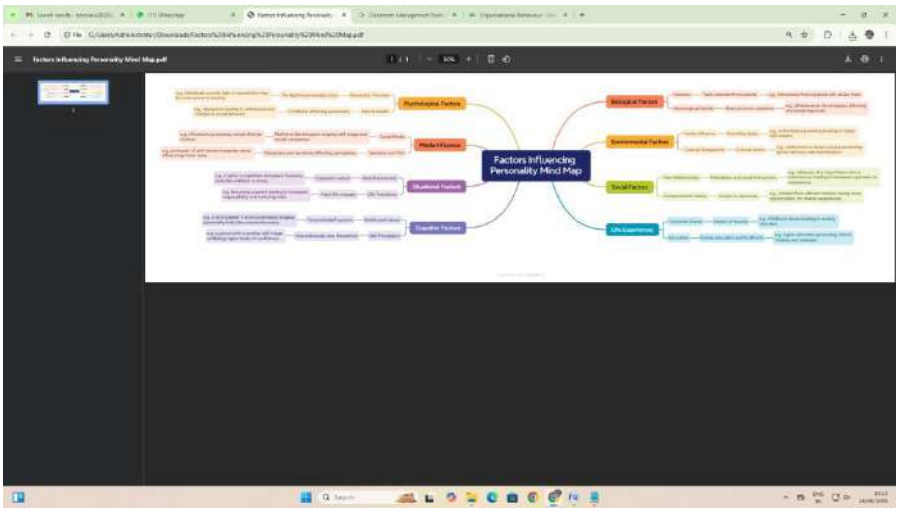
MANAKULA VINAYAGAR
INSTITUTE OF TECHNOLOGY
An Autonomous Institution
Affiliated to Pondicherry University, Approved by AICTE, New Delhi,
Accredited by NBA, New Delhi and NAAC with 'A' Grade
Kalitheerthalkuppam, Puducherry- 605 107.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INNOVATIVE ICT TOOLS EMPLOYED IN CLASSROOMS

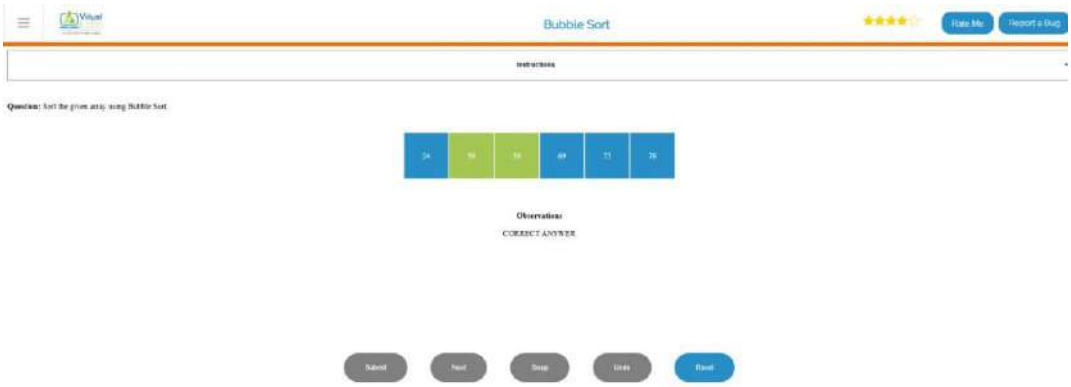
1. **VIRTUAL LAB: 1. https://ds2-iiith.vlabs.ac.in/exp/min-spanning-trees/prims-algorithm/mst_prims_demo.html**

2. **https://ds2-iiith.vlabs.ac.in/exp/min-spanning-trees/kruskals-algorithm/mst_kruskal_demo.html**

Subject Code/Name	**CSPL402/DESIGN AND ANALYSIS OF ALGORITHM**
Year/Sem	**II CSE-B/IV**
Date	
Tool Employed	**VIRTUAL LAB**
Topic	**Preparation of Balance sheet**
Outcome	The Outcome of this Mindmap is to analyse individual and group dynamics within a corporate environment. While each enterprise is distinct and varies across sectors, the common aspiration among them is to foster an organizational culture that harmonizes productivity with employee contentment.
PHOTO	

INNOVATIVE ICT TOOLS EMPLOYED IN CLASSROOMS

DATA STRUVTURE AND ALGORITHMS



INNOVATIVE ICT TOOLS EMPLOYED IN CLASSROOMS

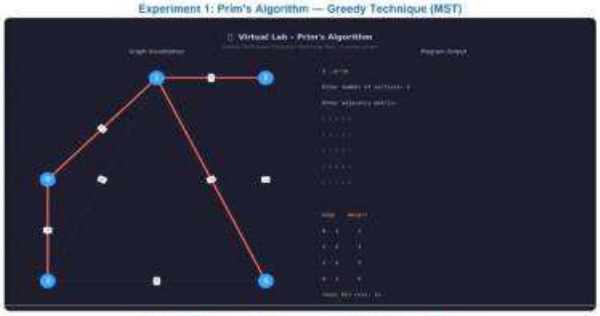
CSPC403/DESIGN AND ANALYSIS OF ALGORITHM

VIRTUAL LABORATORY ASSIGNMENT

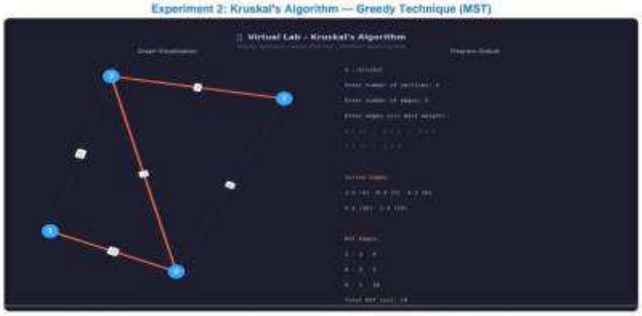
Design and Analysis of Algorithms | Department of Computer Science and Engineering
Academic Year 2025–2026

Virtual Lab Working Screenshots

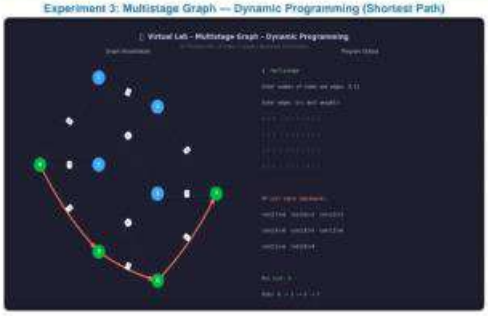
Experiment 1: Prim's Algorithm — Greedy Technique (MST)



Experiment 2: Kruskal's Algorithm — Greedy Technique (MST)



Experiment 3: Multistage Graph — Dynamic Programming (Shortest Path)



Quick Sort Experiment

1.Quick-Sort is:

- a: A stable sorting algorithm
- b: Not a stable sorting algorithm
- c: Depends on the implementation

2. Given a random policy for choosing the pivot, when is the worst case time complexity of Quick Sort hit:

- a: When most of the chosen pivots are the smallest in the subarray
- b: When most of the chosen pivots are the biggest in the subarray
- c: When most of the chosen pivots are smaller than some and bigger than some
- d: a and b

3. Function

```
quick_sort(array) -> sorted-array:
    if array is empty: return array
    pivot = choose_pivot_index()
    smaller_array = [], bigger_array = []
    for i = 1 to sizeof(array):
        if (-----A-----):
            append array[i] to smaller_array
        else if (-----B-----):
            append array[i] to bigger_array
    return quick_sort(smaller_array) + [array[pivot]] + quick_sort(bigger_array)
```

Which of the following pairs can be selected simultaneously for blanks (A) and (B) in the quick-sort algorithm, and would cover all edge cases.

- a: array[i] < array[pivot] and array[i] >= array[pivot]
- b: array[i] <= array[pivot] and array[i] > array[pivot]
- c: array[i] < array[pivot] and array[i] > array[pivot]