

Visualization of Recursion Based Sorting Algorithms



AIM

1. To create tools and notations for visualizing recursion based sorting algorithms and comprehend its dynamics.
2. To integrate these experiments into the existing virtual-labs platform in place of the already existing experiments based on flash.

INTRODUCTION

- Data structures and algorithms form the fundamentals of theoretical computer science.
- The computer science students should have complete understanding of basic algorithms such as sorting, searching, etc as well as data structures for manipulating them.
- The existing visualisation tools are not adequate. On the right side is a screenshot of one of the most famous tools VisualAlgo.net and its visualisation of merge sort.
- Using this we can see the data being sorted but we are unable to make out the recursive structure of the algorithm as well as other details such as time complexity, space complexity etc.
- Using our tool one could make out these parameters as well as write code if necessary without any external sources.

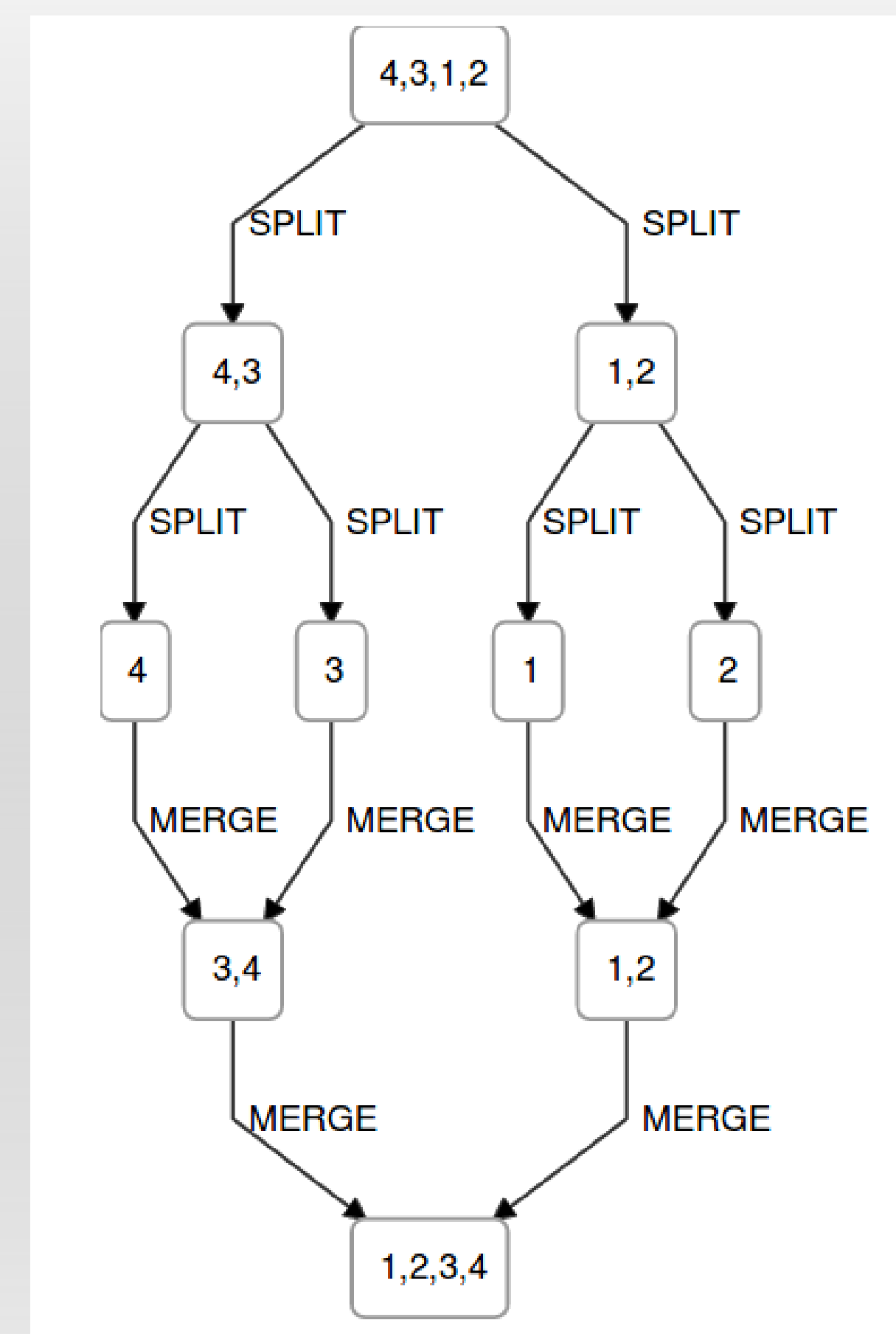
METHODS

The experiments are based on HTML, Javascript and Dagre. The first two components are quite familiar. The main component is Dagre – a d3 based graph visualisation library. It allows us to plot a graph with great flexibility and minimal code. It also offers a great variety of styling and labeling options which made it the most suitable library for this project. These are the screenshots of the actual experiments-

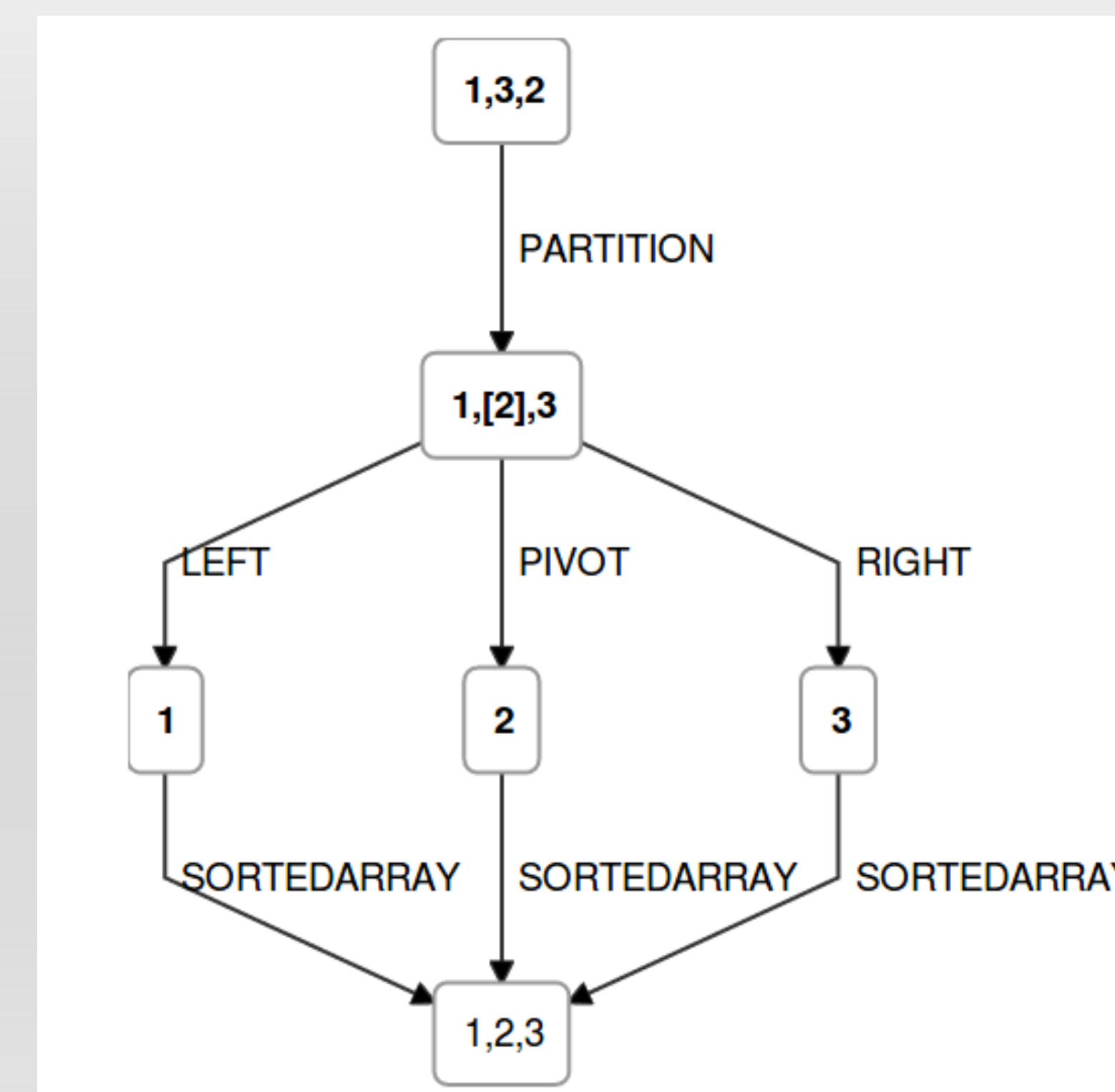
1. Merge sort

The notation of Split and merge has been used. Every merge sort is a black box which when explored gives out a split, 2 similar merge sort on either half and a merge operation.

Merge sort



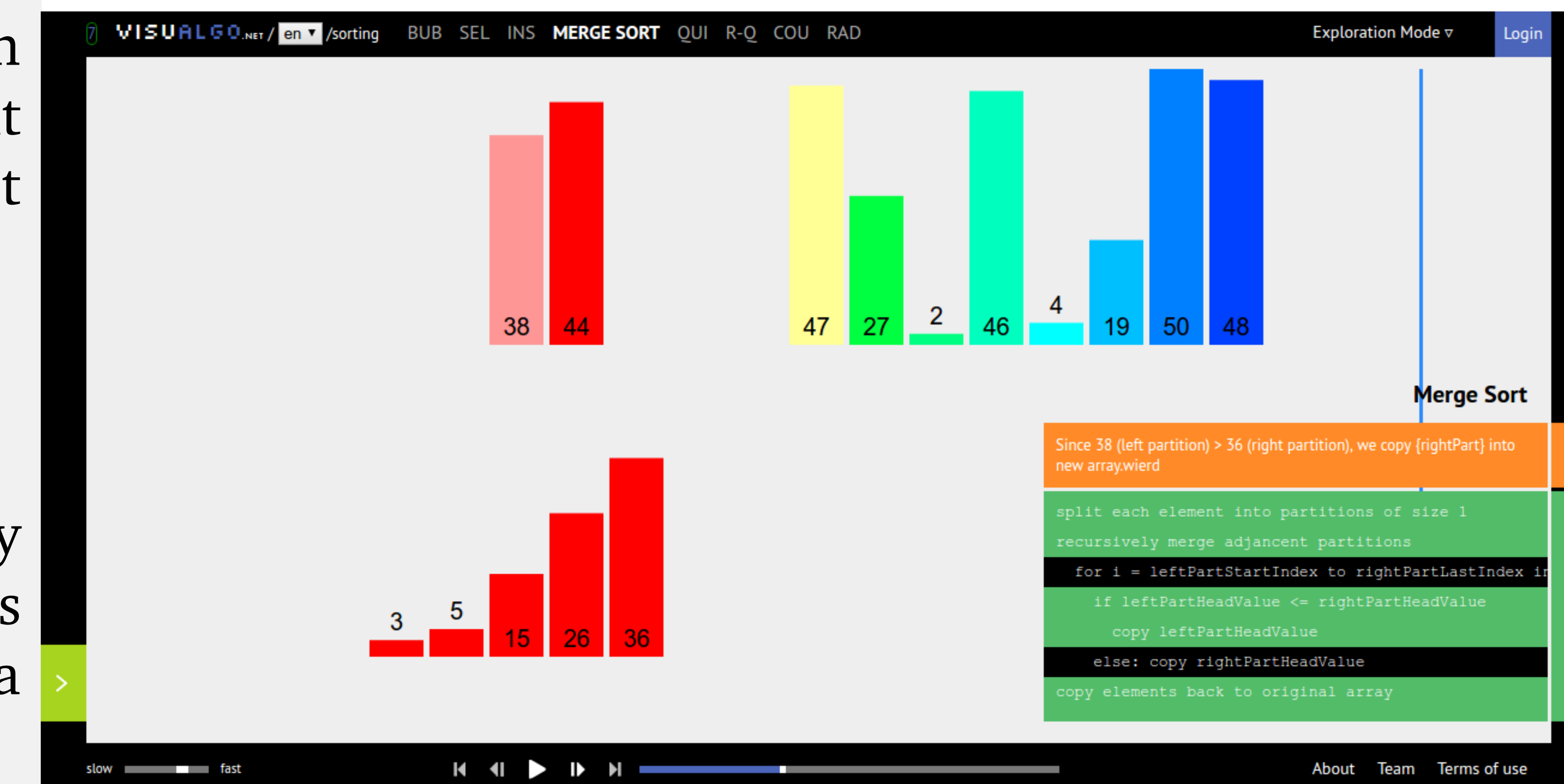
Quick Sort



2. Quick sort

Array is partitioned based on its last elements and elements smaller than pivot are placed left and greater ones are placed right.

VISUAL ALGO



RESULTS

Two of the most common recursion based algorithms i.e. Merge Sort and Quick sort have been converted to javascript and html based experiments.

CONCLUSIONS

1. Algorithms can be visualised using these experiments in a much better way.
2. The code of these algorithms can be written just by using these visualisations.
3. Details such as time-complexity, space complexity etc can be obtained from the visualisation itself.



Intern: Kanay Gupta
Mentor: Venkatesh Chopella

Sources: www.visualalgo.net