



Algorithm Design And Analysis PROJECT

Submitted to :-Dr. Rajiv Mishra

Submitted From :-

Janvi Arora(2k18/SE/070)

Shreoshi Roy (2k18/SE/120)

Introduction

- We have implemented a website which helps us to clearly visualize the algorithms related to searching , sorting and path finding.
- In searching , we have implemented linear search and binary search.
- In sorting algorithms , we have shown heap sort, quick sort , merge sort , insertion sort , selection sort and bubble sort .
- In path Finding , we have used Dijkstra's , A-star, bidirectional , best-first search and depth-first search algorithms .



Tech-Stack Used

For Frontend:-

- HTML
- CSS
- JavaScript
- React

For Backend :-

- Node JS



Searching Visualizer

- We have implemented two searching visualizers. One for linear search and other for binary search . We have used an array of integers . It tells us whether the element we want to search for is present in the array or not .
- Linear search :- In this type of search, a sequential search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.
- Binary search :- Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero.

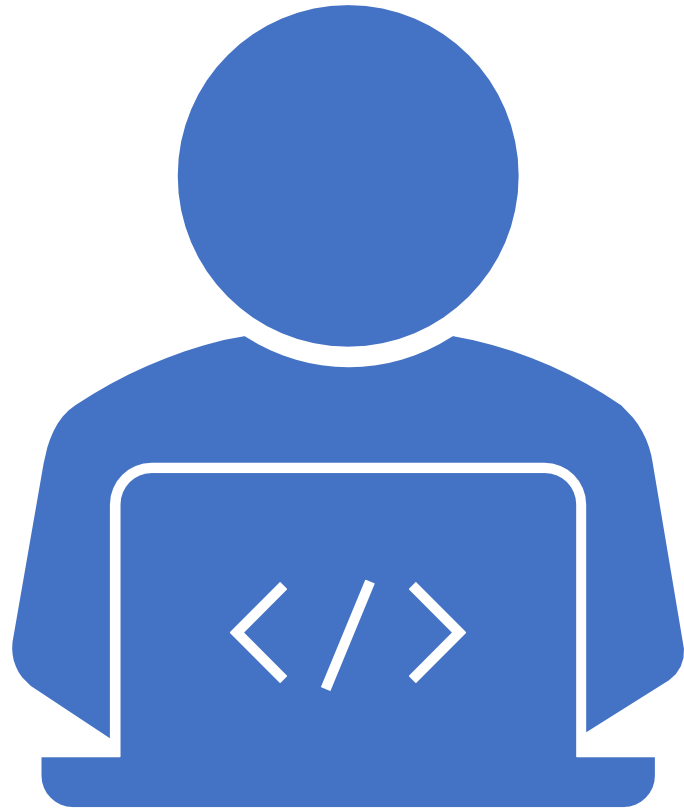


Sorting Visualizer

- The sorting visualizer sorts the array in which light color is assigned the lowest weight and dark color is assigned highest weight .So , it sorts the array from light to dark color .
- Various sorting algorithms we have used are :-
- Heap sort :-Heap sort is a comparison based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. We repeat the same process for the remaining elements.

- Quick Sort :- Quick Sort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quick Sort that pick pivot in different ways.
- Merge Sort :- Like Quick Sort, Merge Sort is a Divide and Conquer algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves.
- Insertion Sort :- This is an in-place comparison-based sorting algorithm. Here, a sub-list is maintained which is always sorted. For example, the lower part of an array is maintained to be sorted. An element which is to be inserted in this sorted sub-list, has to find its appropriate place and then it has to be inserted there.

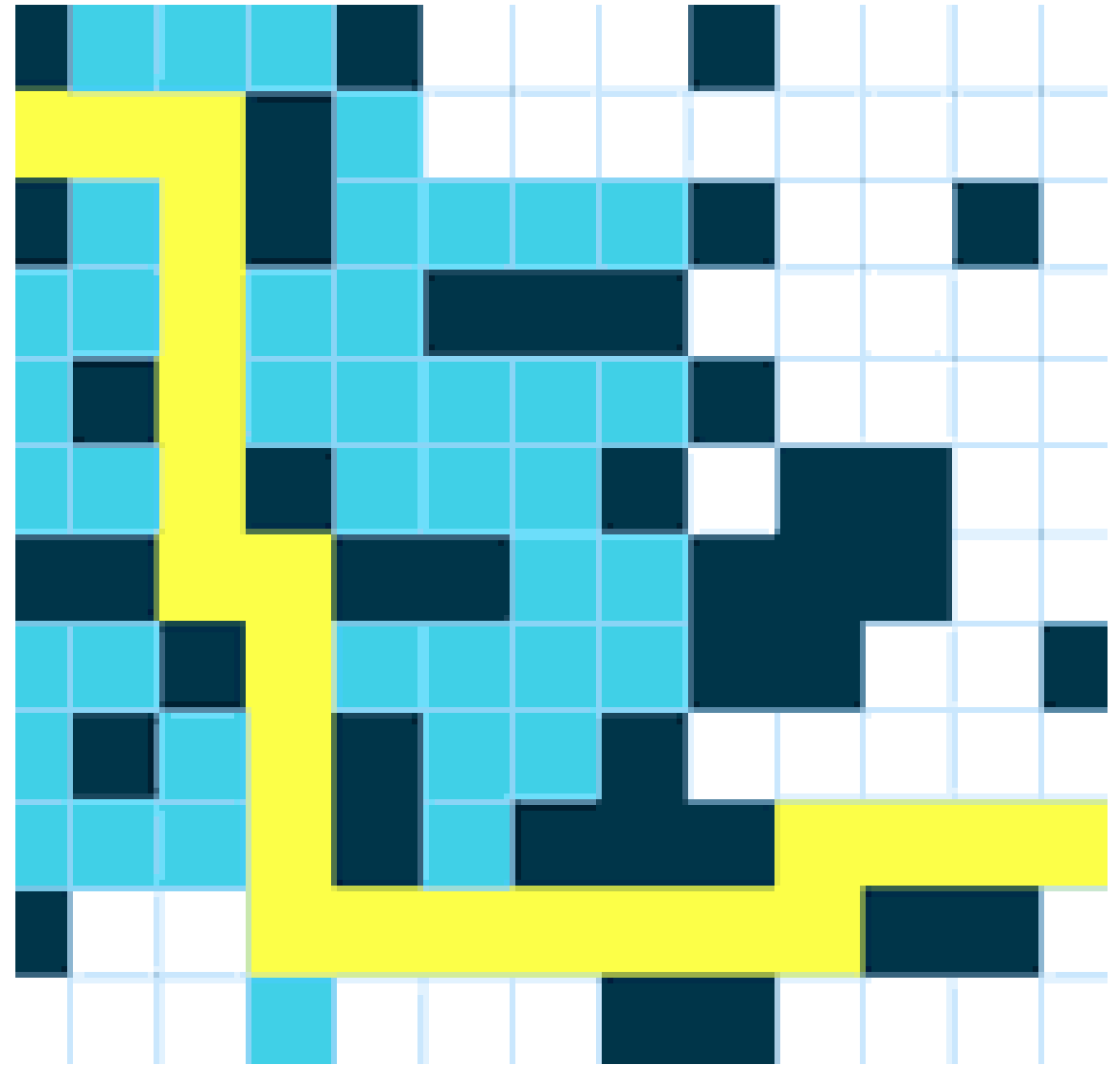


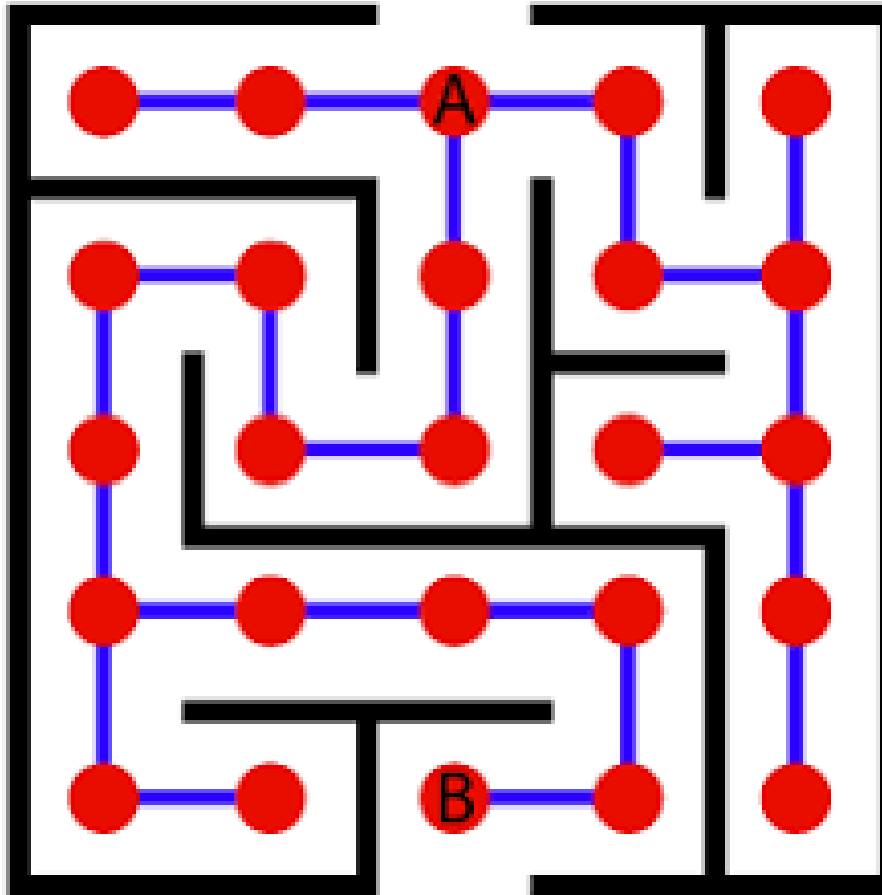


- Selection Sort :-The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array. In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.
- Bubble sort :-Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Path Finding

- Path Finding visualizer allow us to choose the locations of source and destination , place obstacles , generate maze and clear maze .
- The various algorithms used are :-
- Dijkstra's:-For a given source node in the graph, the algorithm finds the shortest path between that node and every other.
- Best first search :-**Breadth-first search (BFS)** is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a search key), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.





- **Depth First Search(DFS)** :-It is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.
- **A* algorithm** :-It is a searching algorithm that searches for the shortest path between the initial and the final state .It has three parameters :-
 - **g** : the cost of moving from the initial cell to the current cell.
 - **h** : also known as the *heuristic value*, it is the **estimated** cost of moving from the current cell to the final cell.
 - **f** : it is the sum of g and h.



Bi-directional search :-

It is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: one forward from the initial state, and one backward from the goal, stopping when the two meet.



Thank You

Janvi Arora(2k18/SE/070)

Shreoshi Roy (2k18/SE/120)