Algorithm Visualizer(AlgoVis)

Kushaal Hulsoor
Priyanka Bhosale

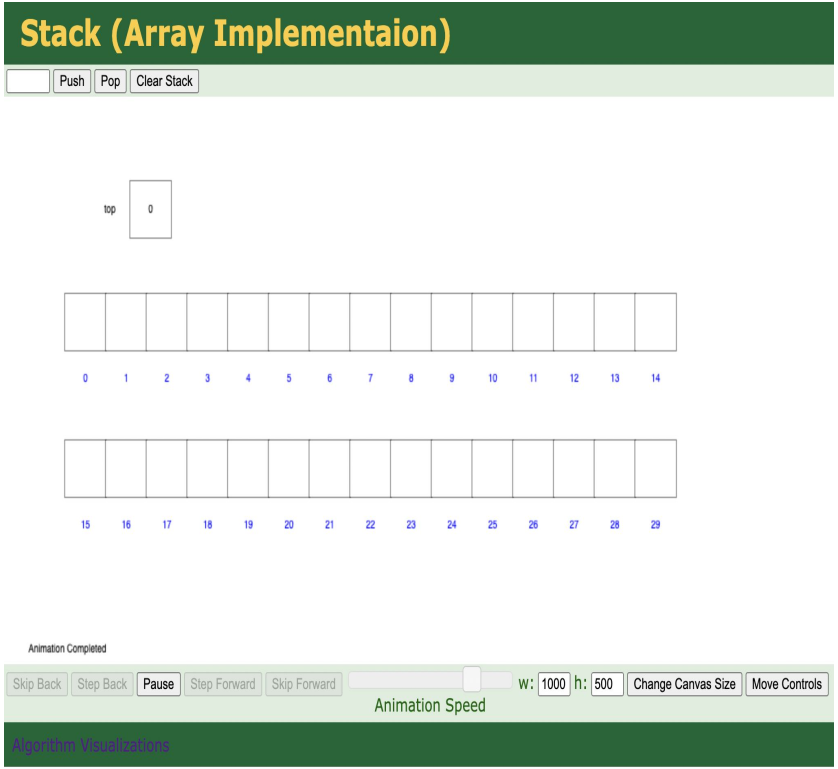
I. Problem And Plan

Description

- Algorithm Visualizer, allows users to visualize different DS algorithms using various animations.
- There are a variety of library functions which will be built on top of javascript for giving graphical effects.

Existing work

- Currently there are multiple existing tools which help visualize different algorithms.[Visualization of Algos]
- Restrictive.
- Focus only on user input values and draw graphs using it.
- User cannot control logic of code as it is developed, maintained and visualized by the developer.

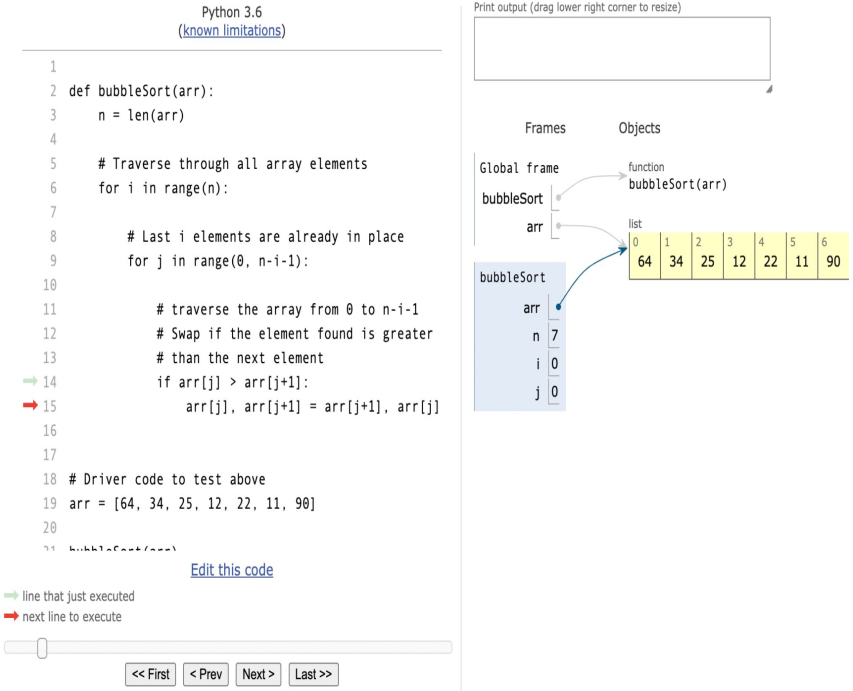


Stack (Array Implementaion)

Animation Completed

Existing work (continued)

- Another example is Python tutor.
- Excellent for knowing control flow execution.
- It tracks the values of variables and displays step execution.
- Doesn’t actually give visualizations of data structures, acts more like a debugger.



Python 3.6
(known limitations)

```
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):

        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]


# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)
```

Edit this code

➡ line that just executed
➡ next line to execute

Print output (drag lower right corner to resize)

Global frame	
bubbleSort	
arr	

bubbleSort	
arr	
n	7
i	0
j	0

function
bubbleSort(arr)

list

0	1	2	3	4	5	6
64	34	25	12	22	11	90

Solution

- Problems mentioned in previous slides can be avoided by creating a generic tool which allows users to write their own algorithms and visualize them.
- The user would be provided with a set of API functions that can be used to build his own logic.
- These functions would be a very basic set of operations that can be applied to a data structure, and calling them would manipulate the data structure in the memory, and also trigger a visual representation of the data changed.

Tools/Frameworks

- Creating a single page web application in ReactJs.
- ReactJs is known for its easy handling of UI component interaction and good performance .
- As the app requires frequent refreshing and creating of new visual components, React was the best choice.

II. DESIGN

V. REFERENCES

References

- https://www.cs.usfca.edu/~galles/visualization/Algorithms.html
  https://www.cs.usfca.edu/~galles/visualization/source.html
    - Copyright 2011 David Galles, University of San Francisco, accessed on Feb 18th, 2022
- https://pythontutor.com/visualize.html#mode=display
- Philip Guo on January 2010, accessed on Feb 18th, 2022
- https://reactjs.org/docs/getting-started.html
    - Copyright © 2022 Meta Platforms, Inc, accessed on Mar 1st, 2022
- https://github.com/unicomputing/s22-algovis
    - Source Code.

References (continued)

- https://algorithm-visualizer.org/
    - https://github.com/algorithm-visualizer
    - https://github.com/algorithm-visualizer/algorithm-visualizer/blob/master/LICENSE
        - Copyright (c) 2019 Jinseo Jason Park, accessed on Feb 18th, 2022
- https://visualgo.net/en
        - Project Leader & Advisor (Jul 2011-present):
            - Dr Steven Halim, Senior Lecturer, School of Computing (SoC), National University of Singapore (NUS)
            - Dr Felix Halim, Senior Software Engineer, Google (Mountain View).
            - accessed on Feb 18th, 2022
            -
- https://cyberzhg.github.io/toolbox/nfa2dfa
    - https://github.com/CyberZHG/toolbox
        - Copyright (C) 2007 Free Software Foundation, Inc.,
    - accessed on Mar 1st, 2022