USER-DEFINED ALGORITHM SIMULATOR

Prof. (Mrs.) J. V. Barpute[1], Siddhant D. Panchal[2], Indira Shinde[3],Sohail Shaik[4]

[1]*Assistant Professor, Department of Computer Engineering, Savitribai Phule Pune University,*
[2]*Pursuing Bachelor of Computer Engineering,* [3]*Pursuing Bachelor of Computer Engineering,* [4]*Pursuing Bachelor of Computer Engineering*
Suman Ramesh Tulsiani Technical Campus Faculty ofEngineering Kamshet, Pune, India.

Abstract:

Learning data structure with static data animation. It is very basic for students of computer science to learn data structures and algorithms but teaching data structures and algorithms is a challenging task for teaching staff by using static animations of predefined data. We are introducing a website tool for allowing students or any programmer to visualize/simulate data structures and user-defined algorithms. Provides complete visualization for the widely used data structure such as array, stack, queue, tree, heap, graph, etc. Provides animation of simple user-defined algorithm.

***Index Terms* - Algorithm animation, user-defined algorithm simulator, data structure visualization.**

1. INTRODUCTION:

The data structure is an important concept in the computer science industry. One challenge that remains in the learning data structure is to understand how data is accessed or used by data structure and using algorithms how data can work. Visualizing how each element to the next, and how a collection of nodes/elements together represent a sequence with a graph or linked-node diagram can help us to understand data structure and algorithm with dynamics of the data elements in the program executions. This project is intended to create an exploration environment, in which students can learn through experimentation. This tool can be used as an effective supplement to the traditional classroom education and textbook for Data Structures and Algorithms courses.

1.1 Objectives

1) To design such a system that can provide a programming environment.
2) Design a system for debugging and simulating algorithms
3) Record all the states/steps

2. RELATED WORK

1.A Tool for Data Structure Visualization and User-Defined Algorithm Animation: -

Tao Chen, Tarek Sobh,[1] has proposed a method based on A Tool for data structure visualization and user - defined algorithm animation In this paper, we present a viewing tool designed to help first-year computer science students study Data Structures and algorithms. This tool not only allows students to visualize commonly used data structures, but also allows readers to write their algorithms in the same Java language - Java My, and watch the creation of algorithms. They believe this tool will be an effective addition to traditional instructions. Due to time constraints, only the majority of commonly used data structures are used in this regard software package versions, including arrays, stacks, lines, binary search trees, lots of binary, essential lines, and indirect graphs. There are two ways to add more data structures visible in this software as a directed graph, weight graph, AVL tree, Dark Red tree, AA tree, splay tree, hash table, etc. One way is by using these data structures in the software. One way would be to develop and implement a software approach package to see user-defined visual data properties and leave the implementation to t e user. This method will allow users to use their visual data properties, which is why they add more flexibility to the software. Another potential future developer for software to highlight the command line of user-defined algorithm files. This will help the user to better track algorithm performance [1].

2) Data Structure Visualization on the Web:

Juan Lin, Hui Zhang,[2] has proposed a method based on Data Structure Visualization on the Web Our main goal is to simplify data comprehension their structure and function. To achieve this, we have suggested an online tool for displaying data structures showing different data structures with their previous graphical shipping. It comes from a complete Python data structure packet, our tool can display invisible data structures without the process of converting the middle to a seamless track is provided during the implementation of the plan. It is necessary to point out that our design shares something similar features of Online Python Tutor mentioned in the previous section. Both tools work in Internet mode, target Python as a visual object, and visualize the internal elements system. The main difference lies in two aspects: the Internet Python Tutor requires input and updates the system later after the assassination. So, any modification will create a duplicate installation and execution process. And it can only manage easily data types, the most complex data types have not yet been set. These two missing parts are full of our tools. Starting with this basic framework, we plan to extend it attacking businesses of more complex data structures with a flexible approach. Animation is also expected to be involved in continuously expressing vague concepts in the data structure as well classical algorithms [2].

3) Visualization of Online Datasets:

Christopher Johnston Downie, Taoxin Peng,[3] has proposed a method based on Visualization of Online Datasets Although, a very promising start, there remains elements that could be performed to improve the tool. Such areas include, implementing additional charting options and increase functionality such as, note taking and animations, provide tutorials and popup prompts to the user to aid education. However, most importantly, Find and implement a method for dynamically detecting data attribute type and chart selection. A proposed further research direction will be via the semantic web and the creation of an appropriate ontology. From which point the tool could be published to the Chrome Store [3].

4) Traversal-based Visualization of Data Structure:

Jeffrey L. Korn Andrew W. Appel [4] has proposed a method based on Traversal-based Visualization of Data Structures We have introduced a new model of software visualization called traversal-based visualization, which is capable of displaying abstract representations of data structures in a debugger. Traversal based visualization makes it possible to write a specification of patterns and actions that provide the semantic information needed to draw objects in an informative way. We have implemented a debugger that allows transformations to be specified with a pattern-based language. The debugger also provides a user interface to this language. The system is functional for Java, and development is ongoing. See the author's home page at http://www.cs.princeton. edu/˜jlk/viz for more information [4].

5) Data Structure and Triangulation Algorithm for Non-Uniform Landscapes:

Yasunori Shiono, Takaaki Goto, Kensei Tsuchida,[5] has proposed a method based on Data Structure and Triangulation Algorithm for Non-Uniform Landscaps, we have proposed a new data structure for rectangular dissection with heterogeneous cells that can be applied in non- uniform landscapes in which the cells have elevation data. The data structure possesses the potential for high-quality expression. Moreover, we have developed a triangulation algorithm for non-uniform terrain visualization. The algorithm generates base triangular meshes for the date structure with consideration of visualization problems. Our approach represents an effective basis for multiresolution models. Future goals of this study are to improve the algorithm for terrain rendering and applied the data structure to other technical fields [5].

4. PROPOSED SYSTEM:

1. All the states are can be recorded. And when client clicks on any recorded state then system will represent states of Data structure.
2. Programming Environment can be provided to handle operations on Data and even large date.
3. Animation's speed can be controlled.
4. Clients can custom inputs as file for large data can be provided.
5. Client can extent the system with the help of extending basic classes.

4.1 Problem statement

The Traditional Way of teaching data structure has been heavily relying on slides assisted with static animations to demonstrate the data operations, which is only useful for pre-designed programs. In Existing DSV tools, when changes are made, they are not recorded and most specifically they didn't provide a programmer environment.

4.2 Architecture Design

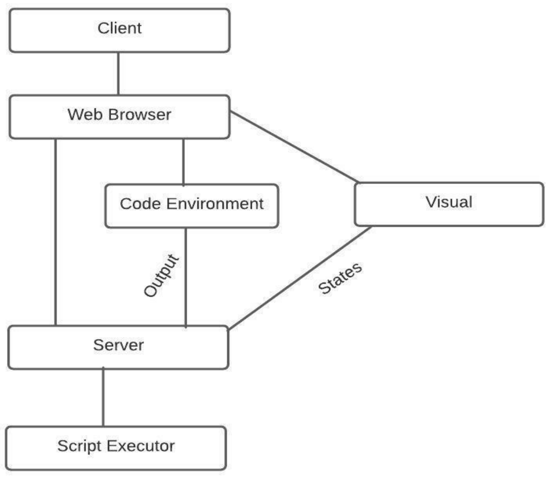

Figure: System Architecture

5. METHODOLOGY:

At the server-side for tracking all the algorithms with data structure written by a programmer, we provided our own data structures by inheriting extending data structure like Array class, Map class to operate on it. After each operation on the data structure, we pushed states to record all the states and then send them to the client. At the client side for simulating/visualizing structure we used D3.jsand some custom animations.

D3 (Data-Driven Document) js:

D3 allows you to bind incorrect data to the Document Object Model (DOM), and then apply the data-driven changes to the document. For example, you can use D3 to generate an HTML table in numerical order. Or, use the same data to create a collaborative SVG bar chart with smoothtransitions and interoperability.

D3 is not a monolithic framework that seeks to provide all the unimaginable features. Instead, D3 solves the root of the problem: effective data manipulation. This avoids proprietary representation and provides unique flexibility, which reflects the full capabilities of web standards such as HTML, SVG, and CSS. With a small overhead, D3 is very fast, supporting large data sets and flexible behavior interaction and animation. The D3 operating system allows for the reuse of code with a variety of official and community-developed modules.

Cytoscape.js:

Cytoscape.js is an open-source graph theory (a.k.a. network) written in JS. You can use Cytoscape.js to analyze graphs and visualization. Cytoscape.js allows you to easily display and manage rich, interactive graphs. Because Cytoscape.js allows the user to interact with the graph and the library allows the client to access user events, Cytoscape.js is easily integrated with your application, especially since Cytoscape.js supports both desktop browsers, such as Chrome, and browsers. mobile phones. as on the iPad. Cytoscape.js covers all the touches you can expect out of the box, including squeezing to zoom, box selection, twisting, and more. Cytoscape.js also has graph analysis in mind: The library contains many useful functions in graph theory. You can use Cytoscape.js head-on in Node.js to analyze the graph in a terminal or web server. Cytoscape.js is an open-source project, and anyone is welcome to contribute. The library is located at the Donnelly Center at the University of Toronto. Replaced Cystoscope Web.

5.1 mathematical model

S=I, E, O
Let S be the proposed system which can be represented as:

1) Input Data I(2)=I1
   I1= Raw Code
2) Intermediate Data: E(2)=E1, E2, E3
   E1=Executed Output
   E2=Executed Error
   E3=Generated States
3) Output Data: O(2)=O1, O2
   O1=Error
   O2=Animated State Simulation.

5.2Algorithm

Algorithm for getting all created object of provided libraries: **-**

Step1**: -** Get the code provided by client.
Step2: - separate all the lines from code.
Step3: - for each line in lines do step 4.
Step4: - check if line consist “\n”, if it is then go to step3 else step5.
Step5: - check if line consist “new” keyword, if it is then going to step6 else step3.
Step6: - store line no in a variable and slice the variable of separator “=” then slice the first one string of separator ““(space) then store the last sliced string.
Step7: - store this line no and variable name from code inside a map and this map apparel to a list and go to step3.
Step8: - Return store list oof map consists all variable name and the line no of each variable name.

6. CONCLUSIONS:

Data Structure View is a tool not only for learning areas but also can be used in the IT industry to analyze data structure and large data sets. Existing systems cannot work with user- defined algorithms, so we propose tools that handle large databases and very specific situations in dynamic data. Visualization is a vital process for student / Researchers /scientists that understand how the thing works. Data structure visualization or in short Algo-visualization tool is a tool that provide some sort of visual animation or simulations on provided data for understanding how the data structure and algorithm work together. This project report on Algorithm visualization provided a more complete understanding and data structure works together on memory. Memory Utilization is also importance thing in algorithm so in this paper we used some algorithm that provides memory optimization.

Another possible future enhancement for the tool is to highlight the executing command line of the user-defined algorithm file. This would help the user to better follow the execution of thealgorithm.

REFERENCES

[1] Tao Chen, taochen@bridgeport.edu ' Tarek Sobh, University of Bridgeport, Department of Computer Science and Engineering, Bridgeport,
CT 06601, sobh@bridgeport.edu

[2] ‘Juan Lin’ juan.lin@louisville.edu ‘HuiZhang Dept. of Comp Sci. & Eng. Speed Schoolof Engineering University of LousivilleLouisville, USA hui.zhang@louisville.edu 978- 1-7281-6251-5/20/$31.00 ©2020 IEEE

[3] Christopher Johnston Downie, Taoxin Peng 10 Colinton Road, Edinburgh, EH10 5DT, United Kingdom cdownie90@gmail.comt.peng@napier.ac.uk 978-1-5090-5756-6/17/$31.00 ©2017 IEEE

[4] Jeffrey L. Korn Andrew W. Appel To appear in IEEE Symposium on Information Visualization (InfoVis ’98), October 1998.

[5] ‘Yasunori Shiono’ Shiono- yasunori-

mb@ynu.jp, ‘Takaaki Goto’ tg@gotolab.net, ‘Kensei Tsuchida’ kensei@toyo.jp, 978-1-5090-5756-6/17/$31.00 ©2017 IEEE SERA 2017, June 7-9, 2017, London, UK

[6] Computing Curricula 2001: Computer Science, Final Report, December 2001. The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery (2002).

[7] L. Naps, J. R. Eagan, and L. L. Norton. JHAVE: An environment to actively engage students in web-based algorithm visualizations. In Proceedings of the SIGCSE Session, pages 109-113, Austin, Texas, Mar. 2000. ACM Press, New York.

[8] Bravo, C., Mendes, A.J., Marcelino, M.J. and Redondo, M.A. Animation and Synchronous Collaboration to Support Programming Learning. Proceedings of Second International Conference on Multimedia, Information and Communication Technologies in Education (m-ICTE’2003), Badajoz, Spain, 2003, pp. 509-513.

[9] Gomes, A.J. and Mendes, A.J. SICAS: Interactive system for algorithm development and simulation. Computers and Education in an Interconnected Society, Kluwer Academic Publishers, 2001, pp. 159-166.

[10] Stasko, T., Badre, A.and Lewis, C. Do Algorithm Animations Assist Learning? An Empirical Study and Analysis. Proceedings of the ACM INTERCHI’93, 1993, pp. 61-63.

[11] John Hamer. A lightweight visualizer for Java. Third Program Visualization Workshop, Warwick, UK, 2004.

[12] Guido Roling and Bernd Freisleben. Animal Script: An extensible scripting language for algorithm animation. SIGCSE Bulletin, 33(1):70-74, 2001.

[13] Ari Korhonen. Visual Algorithm Simulation. PhD thesis, Helsinki University of Technology, 2003. http://lib.hut.fi/Diss/2003/isbn9512267950/isbn9512267950.pdf