GUI for Priority Search Tree

Shuheng Mo

Student ID: 201448174

Supervisor: Konstantinos Tsakalidis

Second Marker: Piotr Krysta

1 Project Description

This project studies the principles of planar orthogonal 3-sided range query and various dynamic data structures such as Cartesian tree, Priority Search tree, and Treap etc. It will implement a dynamic priority search tree that stores a set of 2-d points to support orthogonal 3-sided range reporting queries and insertions/deletions of points. Implement a GUI to the input/output admitting user-commands from the screen.

2 Aims & Objectives

2.1 Aims

This project aims to implement efficient data structures for supporting 2-dimensional orthogonal 3-sided range queries. This project will strive to implement fully dynamic data structures such as a Priority Search tree with general-purpose programming language to support inputting and searching experimental data. An interactive GUI will be constructed for users to customize points' locations in 2-dimensional space and simulating the process of orthogonal 3-sided range queries.

2.2 Objectives

2.2.1 Two-dimensional orthogonal 3-sided range query

In the early stage of the implementation, the developer will attempt to implement data structures to store experimental data and construct static visual two-dimensional data structures which allows the user to insert and delete points (intersecting entries).

2.2.2 GUI implementations

As for the GUI designed GUI for 2-dimensional orthogonal 3-sided range reporting queries, the developer will implement it sequentially as below steps:

1. **Store the points in an array to implement GUI.**
2. **Replace array with the Cartesian tree to support queries and adapt GUI.**
3. **Replace array with PST to support queries and insertions/deletions and adapt GUI.**

3 Key Literature & Background Reading

3.1 Data visualization

Data visualization performs a representation of data. It involves a wide range of graphical forms, such as images reflecting relationships among different datasets, statistical graphics, plots etc. From an information service perspective, data visualization ought to provide users an easy-understanding, efficient interface and therefore benefits users with interacting with graphical data representations and understanding the internal connections with various datasets. Data visualization has several common features: interactivity, multidimensionality, visibility and ease of use [1]. Furthermore, data visualization involves operations on metadata such as data collection, data analysis, and data mining [2].

3.2 Data Mining

Data mining is defined as the process of sorting and classifying a massive amount of data. [3] In the scientific area, data mining was being used to extract required information from data sets with methodologies. Data mining is widely used in statistical analysis, logic analysis of large data sets to contribute to decision-making tasks [4]. For instance, querying the database of a corporation to find the required employees' information, the orthogonal 3-sided range query algorithm can be used to search data sets (entries) under exactly three limited conditions such as salary, age and position. The desired entries will be figured out efficiently within the graphical range as **Fig.1** Shown.

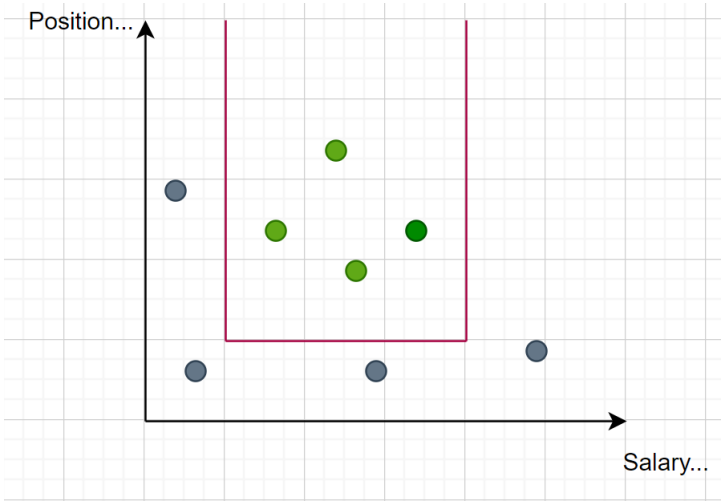


Figure 1: 2-dimensional 3-sided orthogonal range queries

3.3 Data structures for 2-dimensional orthogonal 3-sided range queries

3.3.1 Cartesian Trees

The Cartesian tree is a static data structure thus was designed to support only queries for 2-dimensional orthogonal 3-sided range queries. Cartesian tree can be considered as a binary tree which was heap-ordered (follows the min/max heap property) and efficient in 2-dimensional range searching. As **Fig.2** shown, Cartesian tree was generated from an array deployed in min-heap order, therefore in-order traversal of the tree will be able to output the original number sequence. It takes linear **O(N)** time to build a standard Cartesian Tree and queries specific item in the tree takes **O(NlogN)** time.

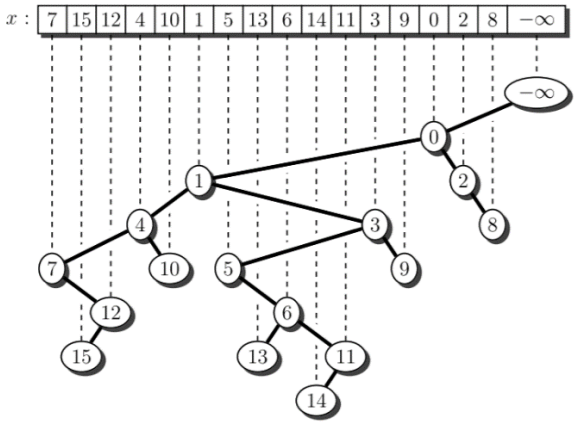


Figure 2: Cartesian Tree

3.3.2 Priority Search Trees

To support queries as in Cartesian Tree and updates (insertions and deletions of points), dynamic data structure priority search tree was introduced. The priority search tree was originally introduced by Edward McCreight, which could be considered as a hybrid data structure that combines the advantages of binary search tree and priority queue [5]. Priority search tree inherited similar properties as priority queue, which it always deletes the item with the highest priority. Formally, a priority search tree is represented in a set **S = {p1, p2, p3 ... }** , which this set can be modeled as points in two-dimensional space with XY-axis such as **p1 (x1, y1)**. On a given range **[l,r]**, Priority search tree can support query to find a specific point **p** such that **p** has the highest priority **p.y** and point **p**'s x-axis lies in the range **[l,r] i.e. l⩽p.x⩽r** . [6] The properties of priority search tree suit requirements of orthogonal search because the min-heap property and queue operations support the insertion and deletion of nodes (points) in this structure rather convenient.

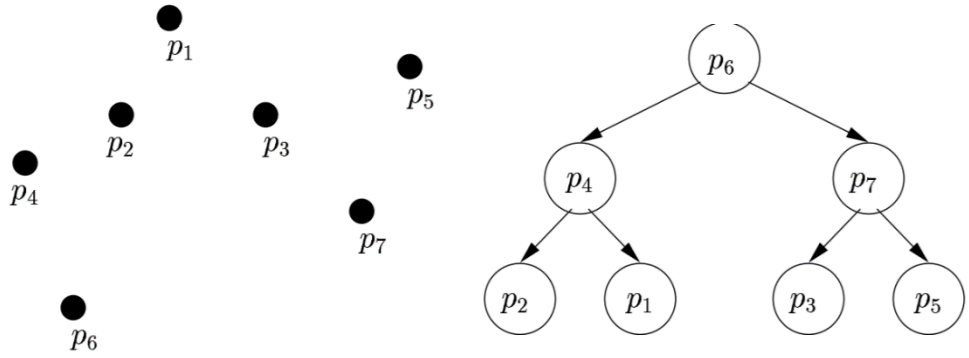


Figure 3: Priority Search Tree

4 Development Process & Method

4.1 Iterative and incremental development

Iterative and incremental development is a software development process that combines both the merits of iterative design and incremental model for development [7]. Regarding the complexity and accuracy of the project, it will be a wise option to develop the system iteratively and implement portions of the system incrementally. By starting with simple implementation, developers will be able to learn from early development and add modifications to the previous prototype.

4.2 C++/C#

C++ is a programming language created by Bjarne Stroustrup as an extension of the C programming language. The language has expanded over time, and modern C++ now has object-oriented, generic, and functional features. Additionally, this language is capable of low-level memory manipulation [8]. C# was developed around 2000 by Microsoft, it is a general-purpose programming language like C++ which possessing generic, object-oriented, and component-oriented programming disciplines components.

4.3 D3.js

D3.js is a JavaScript library for manipulating documents based on data [9]. D3 helps web developers to visualize data using HTML, SVG, and CSS. D3.js's emphasis on web standards provides developers the full capabilities of modern browsers without tying constructing an appropriate framework. In this project, D3.js will be used to design and implement the GUI for this algorithm and visualize the algorithmic processes.

5 Data Sources

Data sets resources used in this project are all artificially created synthetic data sets (points in the two-dimensional plane) rather than open-source datasets from the public platform. Therefore, all the data sources used in this project are permitted to use and legal. The datasets used in the development and test phase will be created regarding the requirement and design document of the project.

6 Testing & Evaluation

6.1 Testing approaches

● Unit Testing

Specify the minimum testable unit of software. A 'unit' could be an individual function, program, or method in procedural programming. By testing these units, the developer will be able to obtain the minimum degree of comprehensiveness desired for the entire system. The steps involving:

1. **Find individual methods in each module/class and test their logic correctness.**
2. **Judge whether the logic has met the requirements.**
3. **Discover flaw in each module and record it.**

● System Integration Testing

System integration testing involving the tests of complete system and all sub-system components. The main purpose of this testing approach is to check the dependencies between different system components. In this test, only intra-system testing with low-level granularity will be applied to fuse modules as a whole. The steps involving:

1. **Apply a bottom-up approach, test the lowest level modules and functions.**
2. **Assemble these modules into a cluster and test the cluster.**
3. **Apply regression testing util bugs/flaws have been detected.**
4. **Test this application/software until all flaws have been revealed.**

6.2 Testing forms (template)

Test plan template							
Test date:		Test Version:		Tester name:		Test component:	
No.	Test Case	Description	Input data	Action	Expected output	Actual output	Success/Fail
1							
2							
3							

7 Ethical Considerations

- **Privacy**

This project requires only artificially created data rather than data from a specific specialized field. Moreover, the completed system requires no login or other user's information thus the project is not harnessing to the user's privacy.

- **Transparency**

All the development processes of this project are open-sourced and can be accessed without extra requirements. The source code and document will be available to users and free to access. Therefore, the transparency of this project was ensured.

- **Business use**

This is a non-profit project which focuses on developing a visualized tool to display the process of 2-dimensional 3-sided orthogonal range query. With the limited functions and capabilities, the outcome of this project will not be used for any business purposes and thus no illegal business issues will occur.

8 BCS Project Criteria

8.1 Fulfill BCS Project Criteria

The entire project studies the principle of the orthogonal range query and strives to implement a graphical user interface for the experiencer to be able to customize the reporting query process. The design and implementation will be carried out originally and innovatively using the knowledge of algorithmics and programming. After the fundamental requirements were achieved, the developer will set appropriate test cases to evaluate the capability of developed items. In the present-day data science field, 3-sided orthogonal query was widely used to explore underlying relations between data entries, to visualize the process of range query will allow the developers and data scientists to efficiently obtain the required data and therefore benefits data mining and decision-making tasks.

8.2 Software and hardware resources

8.2.1 Software

- **IDE: Visual studio 2019 community, Pycharm, HBuilderX etc.**
- **Computing platform: Github etc.**
- **General-purpose programming language: C++, C#, Python, LaTex etc.**
- **Application programming interfaces (API): D3.js etc.**

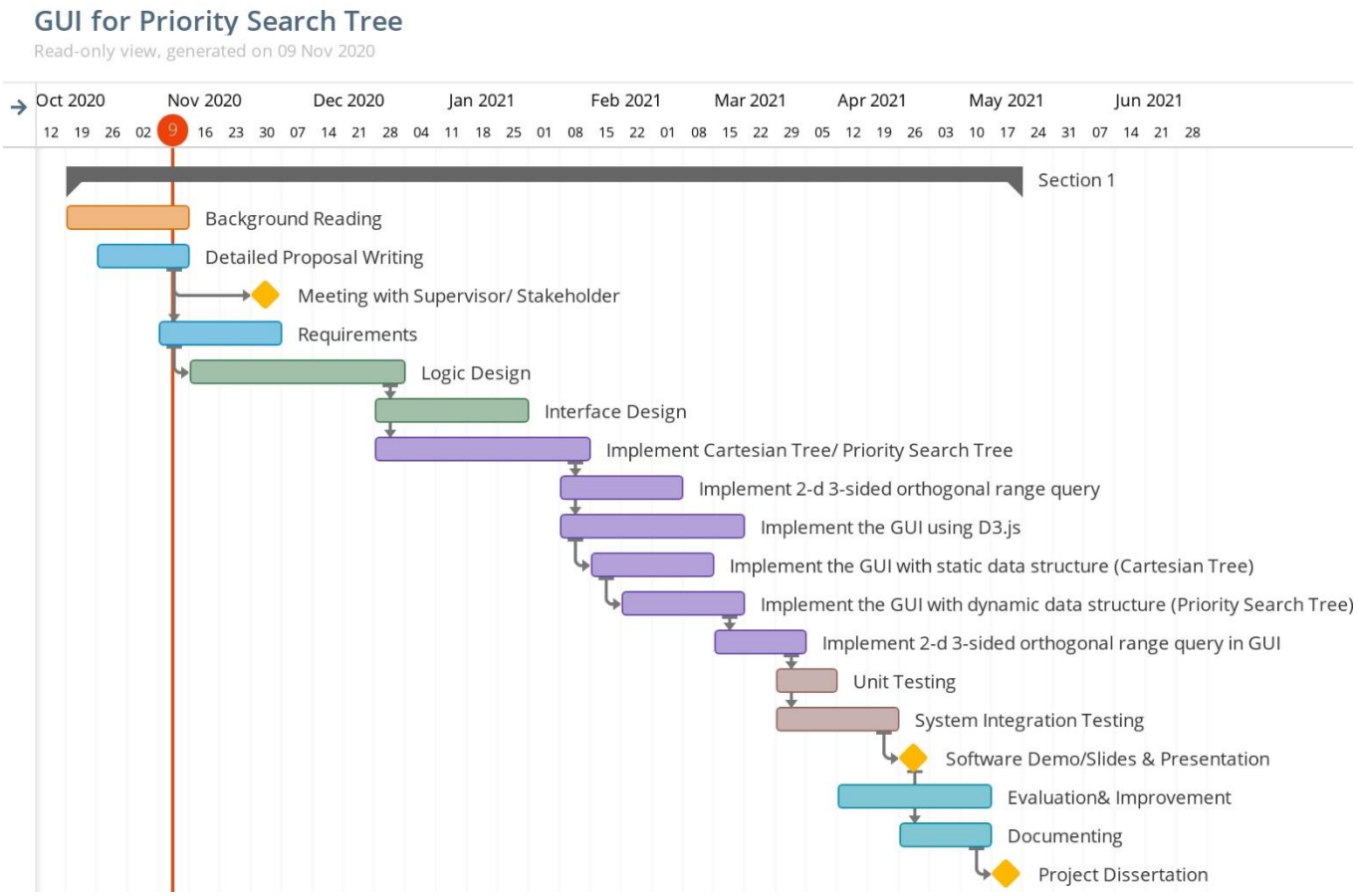
8.2.1 Hardware

Personal Computer (laptop)
Details:

Item	Value
OS Name	Microsoft Windows 10 Pro
Version	10.0.18362 Build 18362
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-53TVJRF
System Manufacturer	ASUSTeK COMPUTER INC.
System Model	N501JW
System Type	x64-based PC
System SKU	ASUS-NotebookSKU
Processor	Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, 2594 Mhz, 4 Core(s), 8 Logical Processor(s)
BIOS Version/Date	American Megatrends Inc. N501JW.209, 24/07/2015
SMBIOS Version	2.7
Embedded Controller Version	255.255
BIOS Mode	UEFI
BaseBoard Manufacturer	ASUSTeK COMPUTER INC.
BaseBoard Product	N501JW
BaseBoard Version	1.0
Platform Role	Mobile
Secure Boot State	On
PCR7 Configuration	Binding Not Possible
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	China
Hardware Abstraction Layer	Version = "10.0.18362.752"
Username	DESKTOP-53TVJRF\asus
Time Zone	China Standard Time
Installed Physical Memory (RAM)	16.0 GB
Total Physical Memory	15.9 GB
Available Physical Memory	7.89 GB
Total Virtual Memory	16.9 GB
Available Virtual Memory	3.83 GB
Page File Space	1.00 GB

The development of this project will possibly involve other hardware devices including portable devices and cloud computing platforms in the future. However, all these devices will be permitted to use and licensed.

9 Project Plan



- **Legend:**

Milestone: Specify the vital events in project process.

Dependencies (arrows between task bar): Specify the subordinate relation of current task to the previous task.

Task bar: Represent tasks in development process, the longer the bar is, the longer duration this task will need. (Planned task durations can be revealed from the top of the graph)

Explanation:

This gantt chart was sketched based on the time arrangement mentioned in COMP390 course overview. It includes the design, development, test and evaluation phases of this project. Tasks of different phases are in different color.

10 Risks & Contingency Plans

Risks	Contingencies	Likelihood	Impact
System failure	Apply the system integration testing method to spot the error and correct it as much as possible.	High	The delay of the project, possibly causes flaws in the final project outcome.
Frequent changes in requirement	Maintain the core requirements and apply iterative development methods, fulfill the requirements bit by bit.	Medium	The final project's outcome varies from the initial design.
High complexity in implementation	Spend more time researching and developing.	Medium	Exhaustion and tiredness. Slow progress.
GUI Corruption	Apply the user interface testing method to find the error and correct it.	Medium	The quality of the graphical user interface will be affected.
Insufficient resources	Find alternatives as soon as possible. Drop useless resources.	Medium	The quality of the final projects' outcome will be affected.
Insufficient communication and cooperation	Re-arrange the schedule to contact the supervisor and classmates more.	Medium	Fail to meet the real requirements of the project.
Lack of time	Re-arrange the schedule to ensure efficient working.	Low	The quality of the project's outcome will be affected.

References

[1] Bai, S. (1) et al. (no date) 'Data Visualization Model Methods and Techniques', IOP Conference Series: Earth and Environmental Science, 252(5). doi: 10.1088/1755-1315/252/5/052063.

[2] Sarkar D. Multivariate Data Visualization with R [J]. Springer, 2008, 25 (b02): 275-276.

[3] Luo Jian. Research and implementation of visual data mining method [D]. Xi'an: University of Electronic Science and technology of China, 2009.

[4] Zhai Xu Jun. Research on Visual Data Mining Technology Based on Parallel Coordinate Method [D]. Beijing: Tsinghua University, 2004.

[5] E. Horowitz, S.Sahi and Anderson-Freed, Fundamentals of Data Structures in C. Computer Science Press, 1993.s

[6] E. M. McCreight, "Priority search trees", SIAM J.Comput., (14,1) 1985, pp. 257-276.

[7] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," in Computer, vol. 36, no. 6, pp. 47-56, June 2003, doi: 10.1109/MC.2003.1204375.

[8]"C++", *En.wikipedia.org*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/C%2B%2B. [Accessed: 31- Oct- 2020]

[9]"d3/d3", *GitHub*, 2020. [Online]. Available: https://github.com/d3/d3/blob/master/API.md. [Accessed: 31- Oct- 2020]