Acceptance Testing

The document displays the user acceptance tests conducted for the user stories implemented so far. Each user story has an acceptance criteria which was collected from the clients and during the tests, the expected outcome was noted. If the expected outcome satisfies all the criteria, the test is a pass else it is a fail.

Version 2.0

This table consists of the user stories for Sprint 1B and Sprint 1C.

Story ID	Tag	Acceptance Criteria	Expected Outcome	Pass /Fail
05	US_05_ADD_BASE_CASES	• Students should be able to visualise how different input arrays affect the structure of a Binary Search tree. • Students should be able to see how a binary search tree is created from a randomised array, a sorted array and a balanced array. • Students should be able to easily switch between the different arrays that are used to build the tree.	• Students can select from 3 different options to configure their input array and it will shuffle the elements in the array to build the desired tree structure • Students will need to build the tree, and playback the algorithm to visualise the tree • Upon clicking the respective checkbox, the input array will change according	PASS
07	US_07_ADD_POINTERS_BST	• Student should be able to see which element in the tree is root and which element is active	• Root and path to active element is marked as blue • Active element is marked as red	PASS
08	US_08_HIGHLIGHT_FOUND_NODES	• Student should be indicated when a node is found or not found after execution of the algorithm	• If such node exist, the node is labelled in red • If such node doesn't exist, a text message should be shown	PASS
09	US_09_SPLIT_TREE	• When parent nodes have one child it should be easily identified whether the node is a left child or a right child.	• Animation splits children nodes to their respective left and right when they are the only child of a parent node	PASS
10	US_10_NEW_NODE_COLLAPSIBLE	• Student should be able to expand and collapse the lines of code pertaining to the creation of a new node	• The lines for creating a new node can be collapsed and expand	PASS
13	US_13_CHOOSE_PIVOT	• Student users should be able to view two options for highlighting the pivot in Quicksort - "Rightmost" and "Median of Three" • On clicking "Rightmost", the pseudocode must highlight the right most element as the pivot. • On clicking "Median of Three", the pseudocode must highlight the median of 3 amongst all the elements as pivot. • Student users must be able to view both the pseudocodes as well as the highlighting of the pivots.	• Two buttons for "Rightmost" and "Median of Three" can be viewed in the parameter panel of Quicksort • On clicking the "Rightmost" button, the pseudocode highlights the right most element as the pivot. • On clicking the "Median of Three" button, the pseudocode highlights the right most element as the pivot. • Student user can only view the pseudocode for rightmost.	FAIL
14	US_14_HIGHLIGHT_PIVOT	• The pivot in Quicksort must be highlighted after it is chosen as it is currently highlighted before the animation is played.	• Pivot of Quicksort is highlighted after it is chosen. • Specifically against the code "Choose Pivot".	PASS
15	US_15_ADD_POINTERS_QS	• Student users should be able to see elements 'i' and 'j' distinctly when the animation is played. • Both the elements must be highlighted to locate them clearly.	• Elements 'i' and 'j' are not highlighted when the animation is played.	FAIL
16	US_16_DISPLAY_SORTED_ARRAY	• Student users should be able to view a sorted version of the input array after the animation has executed completely. • It must be displayed at the bottom of the animation.	• A sorted version of the input array cannot be viewed after the animation has executed completely.	FAIL

21	US_21_DISPLAY_PRIORITY_QUEUE	• Student should be able to view a priority queue for Prim's algorithm. • It must be displayed at the bottom of the animation.	• A priority queue is maintained throughout the algorithm • The priority queue becomes empty at the end	PASS
22	US_22_ADD_FINAL_RESULT	• Student users should be able to view the final result for the "find all nodes" code block. • They must be able to view it without having to expand the code block.	• The final result's animation can be viewed under the code block "find all nodes". • There is no need to expand the code to view this animation.	PASS
23	US_23_ADD_POINTERS_TC	• Student users must be able to locate 'i', 'j' and 'k' elements distinctly in the graph. • They must be highlighted according to the pseudocode.	• The elements 'i', 'j' and 'k' are highlighted in blue in the graph. • The edges (i, k), (k, j) and (i, j) are also highlighted in blue in the graph. • Users can follow the pseudocode with the corresponding highlighting of the elements and edges in the graph.	PASS

Version 1.0

This table consists of the user stories for Sprint 1A.

Story ID	Tag	Acceptance Criteria	Expected Outcome	Pass /Fail
01	US_01_ADD_MODES	• Student users should be able to view Insert and Search as modes for the Binary Search Tree algorithm to differentiate between the two functionalities.	• Labels in the instructions can be viewed as "Insert Mode" and "Search Mode". • Instructions containing steps to be followed for each mode can be viewed.	PASS
02	US_02_LABEL_SPEED_SLIDER	• Student users should be able to view a label named "Speed" next to the speed slider to be able to understand its functionality.	• A label "SPEED" can be viewed on the left of the slider.	PASS
03	US_03_ADD_PROGRESS_BAR	• Student users should be able to view the progress bar next to the play button. • Student users should be able to view the progress of the algorithm in % after clicking the play button. • The progress bar must return to 0% every time a new set of parameters are loaded by the user.	• The progress bar is visible next to the play button. • It loads the progress in % after the play button is clicked. • The bar resets i.e. it returns to 0% after a new set of input parameters are inserted by the user.	PASS
06	US_06_CLOSE_NESTED_BLOCKS	• Student users should not need to collapse a nested block when the respective parent block is collapsed. • The nested blocks must collapse automatically when the parent block is collapsed by the user.	• The nested blocks collapse automatically when a parent block is collapsed. • The user is expected to manually expand the nested blocks again.	PASS
17	US_17_CHANGE_LABELS_FOR_VIEWS	• Student users should be able to view the labels as "Array view" and "Tree view" for the Heapsort algorithm.	• Tree view and Array view are visible for users to view when using the Heapsort algorithm	PASS
18	US_18_LABEL_GRAPH_SIZE	• Student users should be able to clearly identify the buttons for increasing and decreasing the graph size and use them to edit the graph size.	• The button to increase the graph size is labelled as "increase graph size". • The button to decrease the graph size is labelled as "decrease graph size".	PASS
19	US_19_CHANGE_LOAD_BUTTON	• Student users should be able to locate which button can build the graph based on the matrix in control panel easily.	• The button "LOAD" is renamed as "BUILD GRAPH" for easy identification.	PASS
20	US_20_ADD_RESET_BUTTON	• Student users should be able to update the graph if changes are made to the matrix.	• If there exists a built graph, the "BUILD GRAPH" label is automatically renamed as "UPDATE". • The "UPDATE" button can be used to load the graph with updated values in the matrix.	PASS