

5557 Final Exam (20 marks)

(3:30pm – 6:00pm, April 23, 2024)

程序代写代做 CS编程辅导

- Use Eclipse or another IDE.
- Name your project and class FinalYourFirstName.
- You can use the data structure classes from the textbook/slides/class examples or the Oracle API;
- You should submit ALL your the screenshots to Blackboard.



1. (10 marks) In this part, you

- 1) Write a Java method named `buildQueue` which has two parameters for receiving a `Queue` object from the main method and its size ($= 5$). The caller of this method will pass the `Queue` object to this method and have it filled with positive integer numbers. The numbers are 7, 9, 12, 6, 3 which will be added to the `Queue` object in that order; user;
- 2) Write a Java method named `buildAVLTree`. It has a parameter of type `Queue` and another parameter of type `AVLTree`. The caller (main) will pass the `Queue` object built above and an `AVLTree` object to this method and have the `AVLTree` object filled with the five positive integer numbers stored in the above `Queue` object, from head to tail. In the comment section of your Java file, explain what rotations are carried out when each number is inserted into the `AVLTree`;
- 3) Write a Java method named `showAVLTree`. This method has two parameters: one parameter to receive the `AVLTree` object above, the other is a string parameter which will have a value such as “inorder”, “postorder”, “preorder” which specifies how to print the numbers in the `AVLTree` in the specified order;
- 4) Write a Java method named `reverseQueue`. This method has a parameter to receive the `Queue` object alone, together with its size. This method reverses order of the elements in the `Queue` object with the help of a suitable data structure of your choice;
- 5) Test the `buildQueue` method, the `buildAVLTree` method, the `showAVLTree` method and the `reverseQueue` method together in the test driver (main). Take a screenshot of the test output.

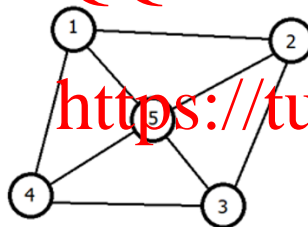
WeChat: cstutorcs

Assignment Project Exam Help

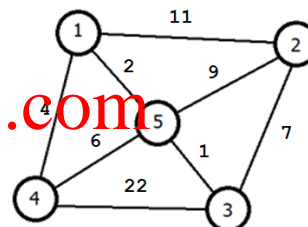
Email: tutorcs@163.com

2. (Total 10 marks)

2.1 Consider the following graphs:



Graph A



Graph B

QQ: 749389476

<https://tutorcs.com>

For the Graph A, do the following:

- 1). Create the vertex array for the graph;
- 2). Create the edge array;
- 3). Create the edge adjacency matrix;
- 4). Build an unweighted graph using the vertex array and the edge array;
- 5). Obtain the DFS search tree and print the search order;
- 6). Obtain the BFS search tree and print the search order;
- 7). Test your code and take a screenshot of the output window;
- 8). If weights are added as shown in **Graph B**, what is the shortest distance from vertex 2 to vertex 4? Use Java code to find it. Test the code and take a screenshot of the output.