

Assignment 10
CSCI 3110: Design and Analysis of Algorithms
Due August 2, 2018

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Assignments are due on the due date before class and have to include this cover page. Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

In this assignment, you are asked to design a data structure. The data structure can be used to store a set S of real numbers. It should support insertions and deletions, that is, you are allowed to add and remove numbers to and from S . Finally, the only query we are interested in is finding the closest pair in S , that is, a pair of numbers (x, y) such that $x, y \in S$ and $|x - y| = \min\{|x' - y'| : x', y' \in S\}$.

As an example, if $S = \{3, 11, 17, 33, 35, 41, 49\}$, then the closest pair is $(33, 35)$. After removing 33 from S , the new closest pair is either $(11, 17)$ or $(35, 41)$. In this case, you only need to report one of them; it doesn't matter which one. After inserting a new element 20, the closest pair becomes $(17, 20)$.

The data structure you construct should support insertions, deletions, and closest-pair queries in $O(\lg n)$ time. The size of the data structure should be linear in the number n of elements in S .

Hint: An (a, b) -tree over the elements in S is the right starting point for your data structure. You need to figure out what additional information you need to store at each node of the tree. The answer should be fairly simple, that is, if you find yourself designing some highly complicated solution, you are on the wrong track.