CS 3100 / In-class Activity 6, Reductions and Worst Case Lower Bounds Proofs

Name	**Computing ID**
Your Name:	

In class: You must work in teams of 2, 3 or 4. Each person writes answers and turns in the sheet at end of class. **Missed class?** Work alone and answer to the best of your ability. Submit to GradeScope by 9am on the 2nd day after in-class activity.

Today, we will prove the following:

> Prove whether there exists a data structure where the operations `INSERT` (which inserts a given element into the data structure), `DELETE` (which removes a given element from the data structure, should it be present), and `FINDMIN` (which returns the minimum element from the data structure) require *O(1)* worst-case time each. The operations only use key-comparisons as their basic operation.

1. Discuss with your team. Do you think such a data structure exists?

2. List some data structures that have some of these properties. For example, `INSERT` into a Linked List is worst-case *O(1)*, since we can insert at the beginning of the list. What about a Priority Queue?

3. Assume we want to prove such a structure does not exist using a reduction. In that case, we need a problem. What problem should we use?

4. Assume the problem we chose in question 3 is **Problem A.** What's the direction of the reduction we want to make for a worst-case lower bound proof? Should we use an algorithm that solves **Problem A** to make an algorithm that uses this data structure to solve another problem (**Problem B**); or should we use this data structure to make an algorithm to solve **Problem A**? Additionally, write down the direction of the reduction: *Problem X reduces to Problem Y* (and stating what X and Y are).

5. Describe the construction of the reduction. How do we take instances of one problem and convert them into instances of the other? How do we get the correct result for the first problem?