CSC 212: Data Structures and Abstractions
Spring 2019
University of Rhode Island

Weekly Problem Set #6

Due Thursday 4/4 at the beginning of class. Please turn in neat, and organized, answers hand-written on standard-sized paper **without any fringe**. At the top of each sheet you hand in, please write your name, and ID.

For the following problems, please refrain from simply copying answers from the lecture slides, as it undercuts the purpose of the assignment.

1. Implement merge sort. Your function should take an array of integers and the indices of the first and last elements in the list to sort.

   ```
   void mergeSort(int arr[], int high, int low)
   ```

2. Implement quicksort. Your function should take an array of integers and the indices of the first and last elements in the list to sort.

   ```
   void quickSort(int arr[], int high, int low)
   ```

3. Define a recurrence relation for the best case for quicksort (perfect partition at every step). Assume that a partition for an array of size n takes n comparisons.
4. Solve your recurrence relation for question 3. Assume T(1) = 1.
5. Define a recurrence relation for the worst case for quicksort (maximally unbalanced partition at each step). Assume that a partition for an array of size n takes n comparisons.
6. Solve your recurrence relation for question 5. Assume T(1) = 1.
7. What are two ways to minimize the chance that worst case occurs? (You may **not** check whether the array is sorted prior to running quicksort).