CSC 212: Data Structures and Abstractions
Spring 2018
University of Rhode Island

Weekly Problem Set #8

Due Thursday 3/29 before 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. The only library you're allowed to use in your answers is `iostream`.

1 Recurrences

1. Find a closed-form equivalent of the following recurrences:

   (a) The Towers of Hanoi:

   $$T(0) = 0; T(n) = 2T(n-1) + 1$$

   (b) The Merge Sort:

   $$T(1) = 1; T(n) = 2T(\frac{n}{2}) + n$$

   (c) Generic:

   $$T(0) = 1; T(n) = T(n-1) + 2^n$$

   (d) Generic:

   $$T(1) = 1; T(n) = T(\frac{n}{3}) + 1$$

2 Merge Sort

1. Determine the running-time of merge sort for a) sorted input; b) reverse-ordered input; c) random input; d) all identical input. Justify your answers.

The following is considered optional.

1. Research and implement Tim Sort. A link about Tim Sort

2. Find a closed-form equivalent of the following recurrence:

   $$f(1) = 3; f(n) = f(\frac{n}{2}) + 1$$

3. Given an array A of size n, find the number of ordered pairs `(i, j)` such that `i < j` and `arr[i] > arr[j]` your answer *must* use Merge Sort.

   Prototype: `int inversions(int* arr, int n)`

   Constraints: `1 <= n <= 1000; 0 <= A[i] <= 1000`

   Example: `{7, 8, 9, 1, 2, 3} -> 9`