**CS 430 – FALL 2023**

**INTRODUCTION TO ALGORITHMS**

**HOMEWORK #6**

1. (6 points) Assume you are creating an array data structure that has a fixed size of n. You want to backup this array after every so many insertion/update operations. Unfortunately, the backup operation is quite expensive, it takes n time to do the backup, regardless of how many items are currently in the data structure. Insertions/updates without a backup just take 1 time unit.

1a) How frequently can you do a backup and still guarantee that the amortized cost of insertion/update is O(1)?

1b) Prove that you can do backups in O(1) amortized time.

2. (7 points) Suppose we wish not only to increment a counter but also to reset it to zero (i.e., make all bits in it 0). Counting the time to examine or modify a bit as THETA(1), show how to implement a counter as an array of bits so that any sequence of n INCREMENT and RESET operations takes time O(n) on an initially zero counter. You must use amortized analysis. (Hint: Keep a pointer to the high-order 1.)

3. (7 points) **Rooted Fibonacci trees** *Tn* are defined recursively in the following way. *T*1 and *T*2 are both the rooted tree consisting of a single vertex, and for *n* = 3*,* 4*, . . . ,* the rooted tree *Tn* is constructed from a root with *Tn*−1 as its left subtree and *Tn*−2 as its right subtree.

3a. Draw the first seven rooted Fibonacci trees.

3b. How many vertices, leaves, and internal vertices does the rooted Fibonacci tree *Tn* have, where *n* is a positive integer? What is its height?

4. (7 points) Give an example of a series of insert() and extract-min operations on a Fibonacci Heap that will yield a heap of n keys with height n-1.

5. (6 points)

|  |  |
| --- | --- |
| Show the data structure that results and the answers returned by the FIND-SET operations in the following program. Use the linked-list representation with the weighted-union heuristic. | A math equations and numbers  Description automatically generated |

6. (7 points) There is an image of  “n by m” pixels. Originally all are white, but then a few black pixels are drawn. You want to determine the size of each white connected component in the final image. Pixels are judged as connected if they share a side.