Advanced Algorithms, Homework 2

Nathan Stouffer

due: 3 September 2020

CSCI 432 Problem 2-1

Collaborators: *n/a*

In this class, we will assign groups for the group project. Now is your chance to weigh in on how we choose them!

1. Describe the problem of choosing the groups formally, including describing what the input and output is. Be sure to explain any properties of the output that are important (e.g., the groups are all of size three and everyone has the same number of characters in their first name).
2. Describe, in paragraph form, the algorithm you propose.
3. Provide this algorithm in the algorithm environment.
4. Prove that your algorithm terminates.

Answer TODO: your answer goes between these lines

CSCI 432 Problem 2-2

Collaborators: *TODO*

Chapter 1, Problem 37 (Largest Complete Subtree).

For this problem, a subtree of a binary tree means any connected subgraph. A binary tree is complete if every internal node has two children, and every leaf has exactly the same depth. Describe and analyze a recursive algorithm to compute the largest complete subtree of a given binary tree. Your algorithm should return both the root and the depth of this subtree.

1. Describe the problem in your own words, including describing what the input and output is..
2. Describe, in paragraph form, the algorithm you propose.
3. Provide this algorithm in the algorithm environment.
4. What is the runtime of your algorithm?
5. Prove that the algorithm is correct.

Answer TODO: your answer goes between these lines

CSCI 432 Problem 2-3

Collaborators: *TODO*

Chapter 1, Problem 9 (Pancakes). When describing your algorithm, please give a prose explanation (in paragraph form) as well as in the algorithm environment. To "Pass", we expect and answer to (a) and (b). To earn a "high pass" on this question, you must answer (c) as well.

Suppose you are given a stack of n pancakes of dierent sizes. You want to sort the pancakes so that smaller pancakes are on top of larger pancakes. The only operation you can perform is a flip—insert a spatula under the top k pancakes, for some integer k between 1 and n, and flip them all over

a. Describe an algorithm to sort an arbitrary stack of n pancakes using $O(n)$ flips. Exactly how many flips does your algorithm perform in the worst case?

b. For every positive integer n, describe a stack of n pancakes that requires $\Omega(n)$ flips to sort.

c. Now suppose one side of each pancake is burned. Describe an algorithm to sort an arbitrary stack of n pancakes, so that the burned side of every pancake is facing down, using $O(n)$ flips. Exactly how many flips does your algorithm perform in the worst case?

Answer

a. We begin with a prose explanation of the algorithm. The input to the algorithm is a permutation of $1, 2, 3, ..., n$ where k references the k^{th} largest pancake. The specific permutation is determined by the stack of pancakes (we order the numbers left to right as the pancakes are ordered top to bottom). Our goal is to produce $1, 2, 3, ..., n$ as output using only the flip operation $O(n)$ times.

   Our algorithm first identifies the largest number k that is out of place, say k is i pancakes from the top. Then we flip the top i pancakes so that k is on top. At this point, we flip the top k pancakes to put k in the correct spot in the stack.

   We now give the pseudocode for this algorithm.

---

stuff

---

b.

c.