COS 423	Theory of Algorithms	Spring 2018

Problem Set 2

This problem set is due Wednesday, February 21 at 11pm via electronic submission. Collaboration is permitted, according to the rules specified in the syllabus.

Read SECTIONS 4.1–4.3 in *Algorithm Design*. Recall that when you are asked to design an algorithm, you must (i) clearly describe the algorithm, (ii) rigorously prove that it is correct, and (iii) analyze its worst-case running time.

1. *(from Google's foo.bar challenge)* Given an integer $n \geq 1$, your goal is to reduce it to 1 by applying some sequence of these three transformations:

   - *Add 1*: $n \leftarrow n + 1$
   - *Subtract 1*: $n \leftarrow n - 1$
   - *Divide by 2*: $n \leftarrow n \div 2$ (This transformation can be applied only if n is even.)

   Design a greedy algorithm that solves the problem in a minimal number of transformations. For example if $n = 29$, one optimal solution is $29 \to 28 \to 14 \to 7 \to 8 \to 4 \to 2 \to 1$; another is $29 \to 30 \to 15 \to 16 \to 8 \to 4 \to 2 \to 1$.

2. You have contracted to complete a set of n computational tasks on a custom-built Tensor Processor Unit (TPU). Unfortunately, your TPU was not working yesterday, so all n tasks are now late. You estimate that task j will take $p_j > 0$ minutes of TPU processing time to complete. Your contract states that you must pay a fine of $w_j > 0$ dollars per minute that task j is late. Design a greedy algorithm to schedule the n tasks on the TPU so as to minimize the sum of the fines.

   The TPU may work on at most one task at a time. Once the TPU starts processing a task, it must complete it before moving on to the next task (i.e., no preemption).

3. Exercise 4.24 in *Algorithm Design* (zero-skew binary tree).