Homework Assignment #3
Due: January 31, 2025 at 5:00 p.m.

1. Suppose we want a *double-ended* priority queue (DEPQ) that stores a multiset Q of keys and supports the following operations.

- INSERT(k) adds k to Q.
- EXTRACTMIN removes one copy of the smallest key from Q and returns it. (If Q is empty, it returns nil.)
- EXTRACTMAX removes one copy of the largest key from Q and returns it. (If Q is empty, it returns nil.)

We decide to do this using *two* binomial heaps *Small* and *Large*, a threshold value T and a variable n that keeps track of the size of Q. Each key of Q is stored in exactly one of *Small* and *Large*. All keys in *Small* are less than or equal to T. All keys in *Large* are greater than or equal to T. The *Small* heap is ordered to support INSERT and EXTRACTMIN, and the *Large* heap is ordered to support INSERT and EXTRACTMAX. Periodically, we rebuild the data structure from scratch so that immediately after the rebuild, keys are split evenly between the two heaps.

[1] **(a)** Since rebuilds are expensive, we want to do them only when they are absolutely necessary. What should trigger a rebuild?

[1] **(b)** Describe how to initialize the data structure if Q is initially empty.

[5] **(c)** Write pseudocode for INSERT and EXTRACTMIN. (The pseudocode for EXTRACTMAX should be symmetric to EXTRACTMIN, so you do not have to give it.) Your pseudocode should be quite short. Add comments to explain anything that is not obvious. Also, it should treat operations on *Small* and *Large* as black boxes: if we replaced the binomial heap implementation of *Small* and *Large* by some other implementation of a priority queue, the algorithms for the DEPQ should still be correct.

[5] **(d)** Use an amortized analysis to give a good upper bound on the total time to do m operations on an initially empty DEPQ. Use big-O notation to state your answer in terms of m. Prove your answer is correct.