1) (10 pts) ANL (Algorithm Analysis)

Consider the task of sorting $\mathbf{n^2}$ integers. Using an insertion sort, this task would take O($\mathbf{n^4}$) time. Using a single heap sort, this task would take O($\mathbf{n^2 lg\ n}$). Consider this hybrid approach and, **with proof, determine its worst case run time, in terms of n.** Assume efficient implementations of each of the heap and linked list operations described. Leave your answer in Big-Oh notation.

1. Separate the $\mathbf{n^2}$ integers into **n** groups of **n** integers each.

2. Create heaps out of each of the **n** groups of integers.

3. Call delete min on each of the **n** heaps, storing these **n** deleted values in a linked list, also storing which heap each value came from.

4. Repeat the following $\mathbf{n^2}$ times:

    a. Loop through the linked list, locating the minimum integer in it, noting which heap it was from. Name the integer **x** and the heap **H**.

    b. Place **x** next in the sorted list and delete it from the linked list.

    c. If **H** isn't empty, delete the minimum item from **H** and add it to the end of the linked list, also storing that the value came from heap **H**.

The first step takes $O(n^2)$ time as each number if processed once.

The run-time of make heap on n integers is O(n). Since we do this n times, the total run-time of step 2 is $O(n^2)$.

Each delete min call on a heap takes O(lg n) time because each of the heaps from which we delete items will have no more than n items. Ultimately, each item from each heap gets deleted, so the total run time of all the delete mins on the heap is $O(n^2 lg\ n)$ time. This encompasses the total run time of steps 3 and 4c.

In step 4, the loop in part (a) takes O(n) time and we repeat it n^2 time, meaning that we spend a total of $O(n^3)$ time on step 4a across all of its iterations.

Step 4b takes O(1) time each time it's executed for a total of $O(n^2)$ time.

If we add up all of these run-times we get:

$O(n^2) + O(n^2) + O(n^2 lg\ n) + O(n^3) + O(n^2) =$ **$O(n^3)$**

An alternate way to do the analysis is to state that step 3 takes O(nlgn) time, followed by a single iteration of step 4 taking O(n) + O(1) + O(lg n) = O(n) time, then noting that since this gets repeated n^2 times, the total run time of step 4 is $O(n^3)$,

Grading: 2 pts for the correct answer
2 pts for explaining why step 2 takes either $O(n^2)$ or $O(n^2 lg\ n)$ time
3 pts for multiplying n^2 times the run time of steps 4a, 4b and 4c.
3 pts for the run time of steps 4a, 4b and 4c as well as step 3 with explanation