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**.