Homework 5 Written

7.12) O(n logn) because though it’s sorted, the heap will still be built in O(n) and though no swapping is needed, which would be negligible at O(1) time for swap, deletemin() still takes O(logn) to percolate the root down no matter the original order.

7.35) a) 2N choose N

b)

7.37) The search can’t be better than O(n log n) and merging takes O(n), therefore:

3log3 + 3log3 + 6 > 6log6, which means that sorting separately and merging takes longer than sorting the list by itself.

Furthermore, this algorithm is also suboptimal because it requires extra memory allocation to store the first 3 numbers, second 3 numbers, and finally the merged group of the two sorted groups. Even if memory allocation was minimized, extra would be needed to transfer contents, and then not much memory is saved. This is similar to the disadvantages most divide and conquer algorithms like mergesort faces.

9.2) If a stack is used for topological sorting, the ordering would be similar to a depth first search, because a stack is first in last out. So for each level of nodes, as a node is popped, its children will be added to the stack but also popped first before other nodes at its parent’s level. A queue would then be breadth search because the nodes are dequeued in the order of first in first out, so children won’t be explored until all nodes on the same level of the parent have been explored. Each data structure may offer a better answer depending on what is wanted, for example an answer by levels or branches.

9.19) If edge weights are bounded, radix sort can be used to give linear complexity of O(n).

9.20) Prim’s algorithm, start with arbitrary vertex and then find most expensive edge connecting this vertex to any other vertex, then keep finding next most expensive edge as long as a cycle is not formed or perform the opposite of Kruskal’s algorhtm, sort all edges of graph decreasing in weight. Add edges in decreasing weight as long as a cycle is not formed. If all the graph/tree is connected or no edges remaining, end of computing MST. This is the same as finding minimum spanning tree, just looking for maximum instead of min weight.