Priority Queues: (15 points)

2) (a) Sue has implemented a priority queue as an array of linked lists. Each list has an enQueue and deQueue method and these methods run in $\Theta(1)$. The enQueue method adds data at the rear of the queue and the deQueue method removes data from the front of the queue. Each array cell holds a pointer to the front of the queue and a pointer to the rear of the queue. There are 5 priorities and so the array has 5 elements. The highest priority is 0 and the lowest is 4. Draw a picture of this data structure after the following data (with priorities shown) has been entered: Mike 4, Sue 3, Tony 4, Jeff 3, Al 4, Ken 3, Noah 0. (5 Points)

(b) Bill has implemented a priority queue as a heap in an array. Draw a picture of the heap as a tree as it is being built. There are 3 priorities. The highest priority (those chosen first) is 2 and the lowest priority is 0. Show each node being entered by redrawing the entire tree after each of the following insertions. You need to decide if this should be a min-heap or a max-heap. (4 points)

Ms. L has priority 0
Mr. B has priority 2
Ms. K has priority 0
Mr. S has priority 2
Ms. D has priority 1
Mr. P has priority 0
Mr. M has priority 0
Ms. C has priority 2

(c) In Bill's priority queue, what is the worse case runtime complexity (Big Theta) of the delete operation? (3 points) $\Theta(\log N)$

(d) In Sue's priority queue, what is the worse case runtime complexity (Big Theta) of the delete operation? (3 Points) $\Theta(1)$