Kristen Orue

Exploration Priority Queues and Binary Heaps:

1. Explain how binary heaps are implemented? Be specific.

Heaps are usually implemented through an array which is where any binary tree can be stored but because a binary heap is guaranteed to be a completed binary tree, it can be implemented through an algorithm that simply compares the parent with the children since a binary heap must be either Min Heap or Max Heap ( Min Heap is where the key at the root must be the minimum value and the Max Heap is the exact reverse) The root element of a Binary Heap is Arr[0]

1. How are items inserted into and removed from a binary heap? Be specific.

When you insert a new node, the new element is first simply added to the end of the heap (as the last element in the array) which most likely will ruin the Heap order property so the next step is the compare the new node with its parent and continuously move the element through the tree with this comparison swapping the parent with the new node depending on whether or not the new node is larger than the parent and whether or not it is a Min Heap or a Max Heap. When deleting an item from the Binary Heap, you not only have to delete the node from the array, you then have to make the tree complete again. You do so by replacing the deleted node with the farthest right now on the lowest level of the binary tree. Then compared the value in the replaced node with its parent node. Filter through either up or down depending on whether the tree is a MinHeap or a MaxHeap

1. Explain the time complexity for a Priority Queue’s enqueue and dequeue operations when implemented as a binary heap (tell me not only what the time complexities are but why they are what they are).

When implemented as a binary heap the time complexity is O(logn) because the comparision is simply between levels. Since Binary Heaps already have a “priority” set in place (Min Heap will be ordered so that the root is the smallest item in the heap while Max Heal will be order to the the root is the largest item in the heap) AND must be a completed binary tree with the most compact N nodes that means it has the smallest possible height which makes it O(logn).