Student name: Ralph NDUWIMANA
Student id: 261066154

Analysis

Big-O Time Analysis

1. Heap sort for a Heap array:

   The time complexity of this algorithm is O(n log n), as it takes O(n log n) time to build the heap, and O(n log n) time to perform the heap sort since there are n elements in the array.

2. Insertion sort for a Priority Q with sorted Q**:**

   The time complexity of insertion sort for a priority queue with a sorted queue is O(n). This is because each element is only compared to the element before it, and there are n elements in the array.

3. Selection sort for a Priority Q with not sorted Q**:**

   The time complexity of this algorithm is O(n^2), as it takes O(n^2) time to find the minimum element and swap it with the first element, then find the second minimum element and swap it with the second element, and so on.

   The time complexity of selection sort for a priority queue with a not sorted queue is O(n^2). This is because each element is compared to every other element in the array, and there are n elements in the array.

Can you comment which one is faster?

Heap sort is faster than insertion sort and selection sort.
The insertion sort is faster than the selection sort, as it has a lower time complexity.

Do you think if we use array or LinkedList, we could speed the program up?

Array would be faster than LinkedList.

Does the array initial order have any effect on the speed?

No, the initial order of the array does not affect the time complexity of the algorithms.