1.

heap:nlogn

cause its binary tree,assume there are n nodes, then the height is logn, which means even you upheap/downheap from root to outer leaf, it will just take swap logn time while swap is O(1), and there are n nodes to swap ,so logn \*1 \* n=nlogn

sorted arraylist n^2

because it sorted arraylist pq, we must take O(n) time to find where to insert and shift,and we must insert n times, so it is n\*n=n^2

2.

heap:1

sorted arraylist 1

they are both based on arraylist in fact so when we swap, we don’t need assistant ADT,

for the space to build heap and sorted arraylist, we need O(n) space because they are arraylist.

3.heap is quicker especially when n is large

because O(n) of heap is nlogn

O(n) of sorted arraylist is n^2

nlogn<n^2