Gray Houston

ghousto

Project 2

Analysis Questions

1. Both the best case and the worst case running time is O(nd). To start, the inner loop will execute in O(d) time, as the program loops over the next d elements to find the smallest element to insert. Next, the outer loop will iterate n times and therefore run in O(n) time. Because the inner loop will run proportionally to the outer loop, the overall running time is O(nd). Because the inner loop will always run in O(d) time (and can never run for shorter), both the best case and the worst case will be O(nd) time.
2. The best and worst case running time is O(nd logd). To start, the creation of each small min-heap will take log(d) because it uses the bottom-up heapify method and there will be d heaps created. Therefore, creating all of the heaps will run in O(d logd). Deleting each element from a heap will run in O(n) time. Therefore, the algorithm will run at O(nd logd) time. This implementation runs in this time because it is in-place.
3. If T(n) is the time that it takes for my implementation of Mergesort to run, then

T(n) = 2 \* T(n/2) + K where K is some constant locality value

= 22 \* T(n/22) + K(1 + 2)

= 23 \* T(n/23) + K(1 + 2 + 22)

…= 2K \* T(n/2K) + K(1 + 2 + 22 + … + 2K-1)

= 2K \* T(n/2K) + K \* n

Therefore, when d is set to a constant value, my Mergesort will have an asymptotic O(n) running time.

1. A
2. A
3. For an array of size n, when d=0, that means that each element is at most 0 indexes away from its final position. Where there are 0 indices between an element and its final position, that means it has to be in its final position. Therefore, d=0 for