Problem 4.1 Merge Sort

a)

merge sort.cpp

b)



c)

The variant k represents how many elements need to be sorted using Insertion Sort before Merge Sort.

For the Best Case, whatever k is changed into, the time complexity does not change because it is sorted already, no need to go through any sort process.

For the Worst and Average cases, the higher k is, the more time it takes for the algorithm to execute. This is due to the number of elements being sorted using Insertion Sort, which is slower than Merge Sort, therefore, as k increases, less elements are sorted using Merge Sort which leads to more time.

Thus, as k goes bigger, the asymptotic time complexity is going to be more like Insertion Sort instead of Merge Sort.

Problem 4.2

a)

图片包含 文字, 白板

描述已自动生成

b)

![图片包含 文字, 白板

描述已自动生成]()

c)

图片包含 文字, 白板

描述已自动生成

d)

According to the recursion tree, 2^n has occupied the entire sequence.

Thus this answer is T(n)= θ(2^n)

图片包含 文字, 白板

描述已自动生成