Selection Sort

List Size	Comparisons	Time (seconds)
1,000 (observed)		
2,000 (observed)		
4,000 (observed)		
8,000 (observed)		
16,000 (observed)		
32,000 (observed)		
100,000 (estimated)		
500,000 (estimated)		
1,000,000 (estimated)		
10,000,000 (estimated)		

Insertion Sort

List Size	Comparisons	Time (seconds)
1,000 (observed)		
2,000 (observed)		
4,000 (observed)		
8,000 (observed)		
16,000 (observed)		
32,000 (observed)		
100,000 (estimated)		
500,000 (estimated)		
1,000,000 (estimated)		
10,000,000 (estimated)		

Merge Sort

List Size	Comparisons	Time (seconds)
1,000 (observed)		
2,000 (observed)		
4,000 (observed)		
8,000 (observed)		
16,000 (observed)		
32,000 (observed)		
100,000 (estimated)		
500,000 (estimated)		
1,000,000 (estimated)		
10,000,000 (estimated)		

1. Comparing the sorts at a general level, is one sort *always* better than the others?

2. Which sort is better when sorting a list that is already sorted (or mostly sorted)?

3. You probably found that insertion sort has about half as many comparisons as selection sort. Why?

4. Given the above observation, why are the times for insertion sort not half what they are for selection sort? (For part of the answer, think about what insertion sort has to do more of compared to selection sort.)