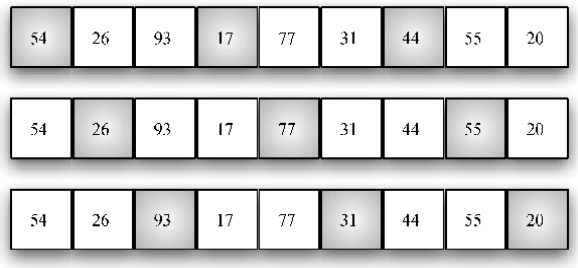
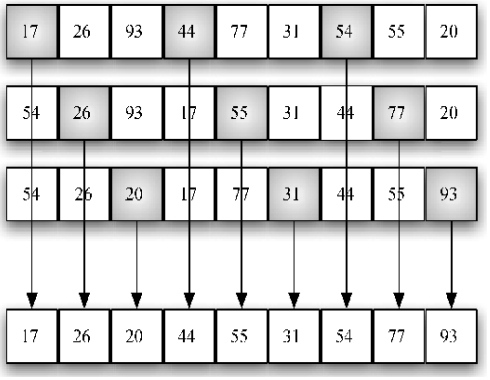
DSA Assignment 1 Write Up by LIUJIAJUN 211283E

In this assignment 1, I added Shell Sort and Cocktail Sort to sort No of Pax and Customer Name respectively in bonus features. Next, I will introduce my understanding of these two algorithms in detail.

Shell Sort:

It improves on insertion sort by dividing the list into sublists and applying insertion sort to each sublist. How to split the list is the key to shelll sorting - not continuous splitting, but using increments to select all elements of the interval to form sublists.
For example, the list in the figure below has 9 elements. If the increment is 3, there are 3 sublists, each of which can apply insertion sort, although the list is still not completely ordered, by sorting the sublists, we have brought the elements closer to their final positions





Although the last step requires an insertion sort, the list is already preprocessed by incremental insertion sort, so the last step of insertion sort does not require multiple comparisons or moves. That is, each round of traversal produces a "more ordered" list, which makes the last step very efficient. So in this example only requires 4 more moves. The time complexity of shell sort is roughly between O(n) and O(n^2). In my assignment, the time complexity is O (n^2).

Cocktail Sort

It is also known as "bidirectional bubble sorting". It is a variant of bubble sort. This algorithm differs from bubble sort in that it sorts the sequence in both directions.
(1) First bubble sort the array from left to right (ascending order), then the largest element goes to the rightmost end;
(2) Then bubble sort the array from right to left (descending order), then the smallest element goes to the leftmost;
And so on, changing the direction of bubbling in turn, and continuously shrinking the range of unsorted elements until the end of the last element.
Time complexity, two loops, time complexity is O(n^2)

So why is it that cocktail sorting is better than bubble sorting?

Consider a sequence like this: (2,3,4,5,1). If use cocktail sorting, it can do it in one back and forth, and bubble sorting requires four runs.
The reason is that the bubbling is only in one directon. If it bubbles from left to right, it will be not good for the small number at the back of the array as it will require more swap. Vice versa for right to left with big number. The advantage of cocktail sorting is that the two worst situations can be avoided, thus bringing some improvements in performance.