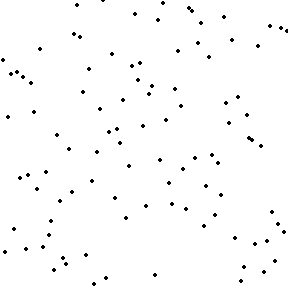
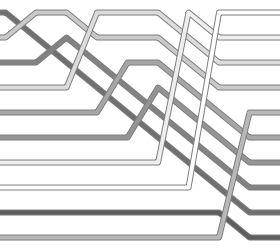
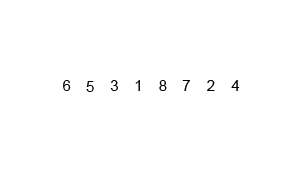
Sorting Algorithms Time Complexity

When we consider running time complexity with sorting certain algorithms, we need to associate with the what kind of data structures we can use in certain scenario for each sorts.

In Selection Sort, we use O(n^2) time complexity for a best and worst case in order to make it inefficient to large list for further simplicity to perform better over complicated algorithms in certain cases. The algorithm has two structures: the sub-list of items already sorted out and the sub-list of items remained to be sorted out for the rest of the list. This describes that this kind of sorting algorithm provides a linear and yet limited approach to data structure compared to insertion sort. 

Like selection sort, we use O(n^2) for a worst case, but use O(n) for best case scenario for Insertion Sort in time complexity in order to build the final sorted array one item at a time. It provides several advantages to small data sets like quadratic sorting algorithms for a further simple implementation and optimization of the algorithm compared to other kinds of sort. 

Merge Sort is one of those complicated sort that is considered more efficient for input order and equal elements for a sorted output. In this case, we use the O(n log n) time complexity for best and worst case performance on the algorithm. While using a merge sort algorithm, we are dividing and merging with elements in order to sorted it out correctly. It is also called the divide and conquer algorithm. 

At last, we have the Quick Sort algorithm. When it comes to performing, it uses O(n log n) for best, but O(n^2) time complexity for worst case scenario. This algorithm is another complicated sorting algorithm which is also efficient like merge sort in order to place the elements of the array in order in a systematic way. Unlike merge sort, quick sort helps us to compare elements in a relation and relative to other elements in the array and yet it is implemented two or three times faster than merge sort. 