1 Sort comparison

Sort algorithms are described very well in plenty of sources. Therefore they will not be described here. The only thing that is provided here is simple comparison of few, most recognizable sorting algorithms along with some extra comments.

Algorithm Name	**Complexity (average)**	**Space Complexity**	**Comment**
Bubble sort	$O(n^2)$	$O(1)$	Time complexity depends on input data
Selection sort	$O(n^2)$	$O(1)$	
Insertion sort	$O(n^2)$	$O(1)$	Time complexity depends on input data
Merge sort	$O(nlogn)$	$O(n)$	There are implemention versions of in-place, which reduce space complexity
Quick sort	$O(nlogn)$	$O(n)$	Time complexity depends on input data and selection of starting point
Heap sort	$O(nlogn)$	$O(n)$	

Comments:

- Some algorithms are sorting algorithms "in-place", which basically means that they do not use extra memory to save some partial results. These algorithms have therefore space complexity of $O(1)$.
- Some algorithms have different complexity when input data has some special shape. E.g. selection sort will run faster if delivered data is already sorted (complexity $O(n)$).
- Quick sort is untypical algorithm, where complexity depends not only on shape of input data (sorted, unsorted, all elements are same) but also on value of selected pivot point. Therefore there are different techniques to select one, like:
  - Selection of random value in fragment
  - Selection of middle value in fragment
  - Selection of median of first, middle and last value in fragment