|  |  |
| --- | --- |
| **Insertion sort**   * Insertion sort is a simple sorting algorithm. * Insertion is expensive, requiring shifting all elements over by one. * Worst case time: O(n2) * It is Adaptive: O(n) time when nearly sorted * Number of swaps is depended on the amount of input * Space requirement  O(n2), O(n) in place and O(1) extra space * Suitable for small files | **Quick sort**   * Quicksort is a divide and conquer algorithm * It relies on a partition operation. * Worst case time: O(n2) * Adaptive for sorting in the random-access store of a device. It depends both on the input size and the pre-sort in the input * Expected number of swaps is 1/6 the expected number of comparison (1.4n log2n) also depends on the specific input vector. * Space requirement O (1) in place and O (log(n)) extra space. * Suitable for large files |

Task #2: Picking the right sort of sort

I would recommend quick sort because it can conquer algorithm that is relatively efficient for large lists, whereas insertion sort is better for simple sorting and small lists which are mostly sorted. Quick sort overcomes other characters. Quick sort is much faster than insertion sort. Quick sort can be done in linear time and in place. In quick sort, worst case can be avoided however, the same cannot be done in insertion sort. Insertion sort is also more expensive than quick sort.Insertion sort works by taking each element one by one, inserting them in their correct position and quick sort works by choosing a random pivot point and separating the greater and smaller values on either sides of the pivot point.