DESIGN AND ANALYSIS OF ALGORITHMS

PROJECT

NAME : SHUBHAM SHANKAR
STUDENT ID : 1001761068

Sorting Algorithms :

A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure.

This project will display the important properties of different sorting techniques including their complexity, stability and memory constraints.

DIFFERENT SORTING ALGORITHMS USED:

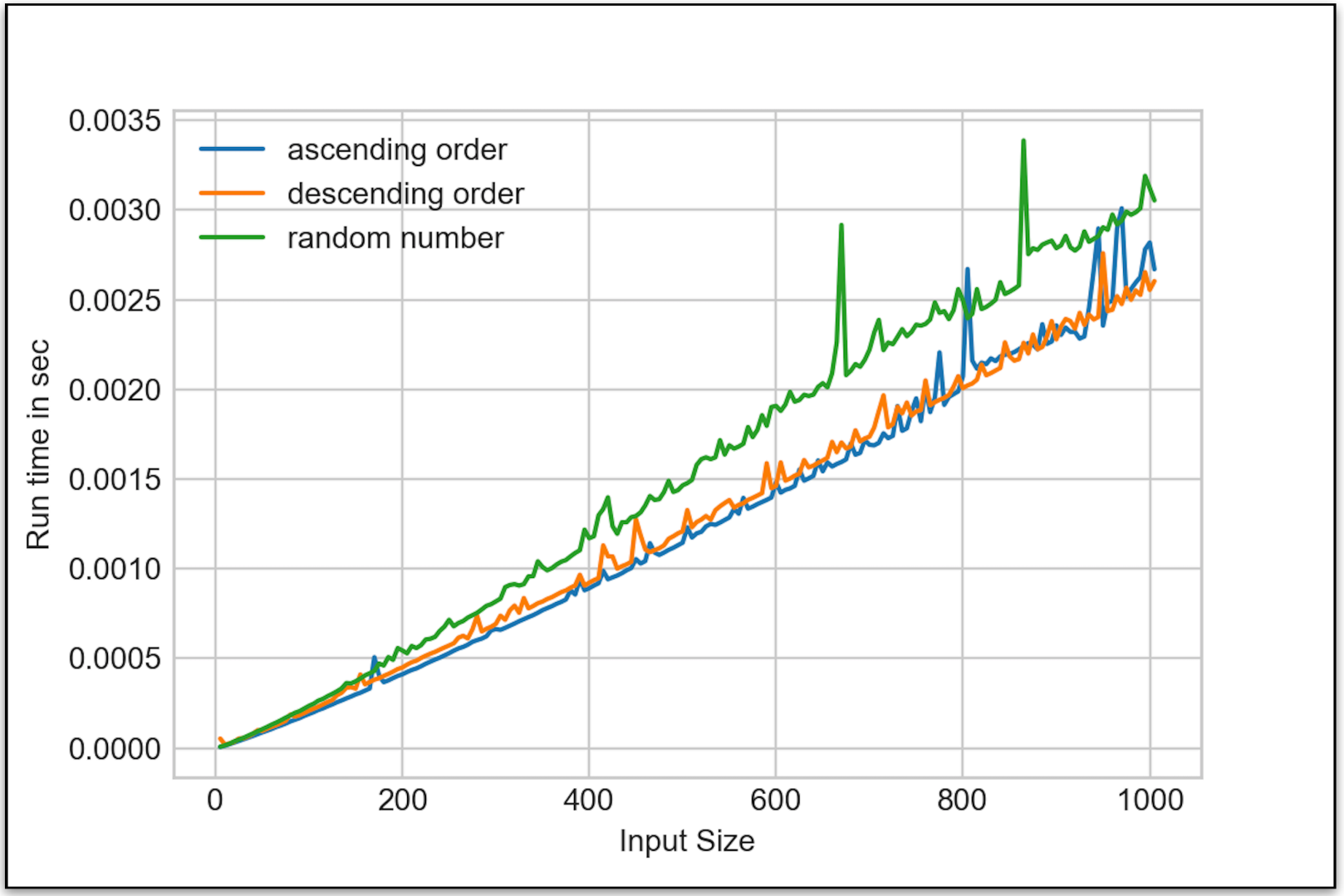
- Mergesort
- Heapsort
- Quicksort (Using median of 3)
- Insertion sort
- Bubble sort

MERGE SORT:

- Merge Sort is a divide and conquer algorithm.
- It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves.
- The merge() function is used for merging two halves.
- The merge() is key process that assumes that array are sorted and merges the two sorted sub-arrays into one.
- BEST CASE , WORST CASE AND THE AVERAGE CASE OF MERGE SORT IS THE SAME.

O(n log n)

RUN TIME FOR DIFFERENT INPUT SIZE IS PLOTTED BELOW FOR MERGE SORT

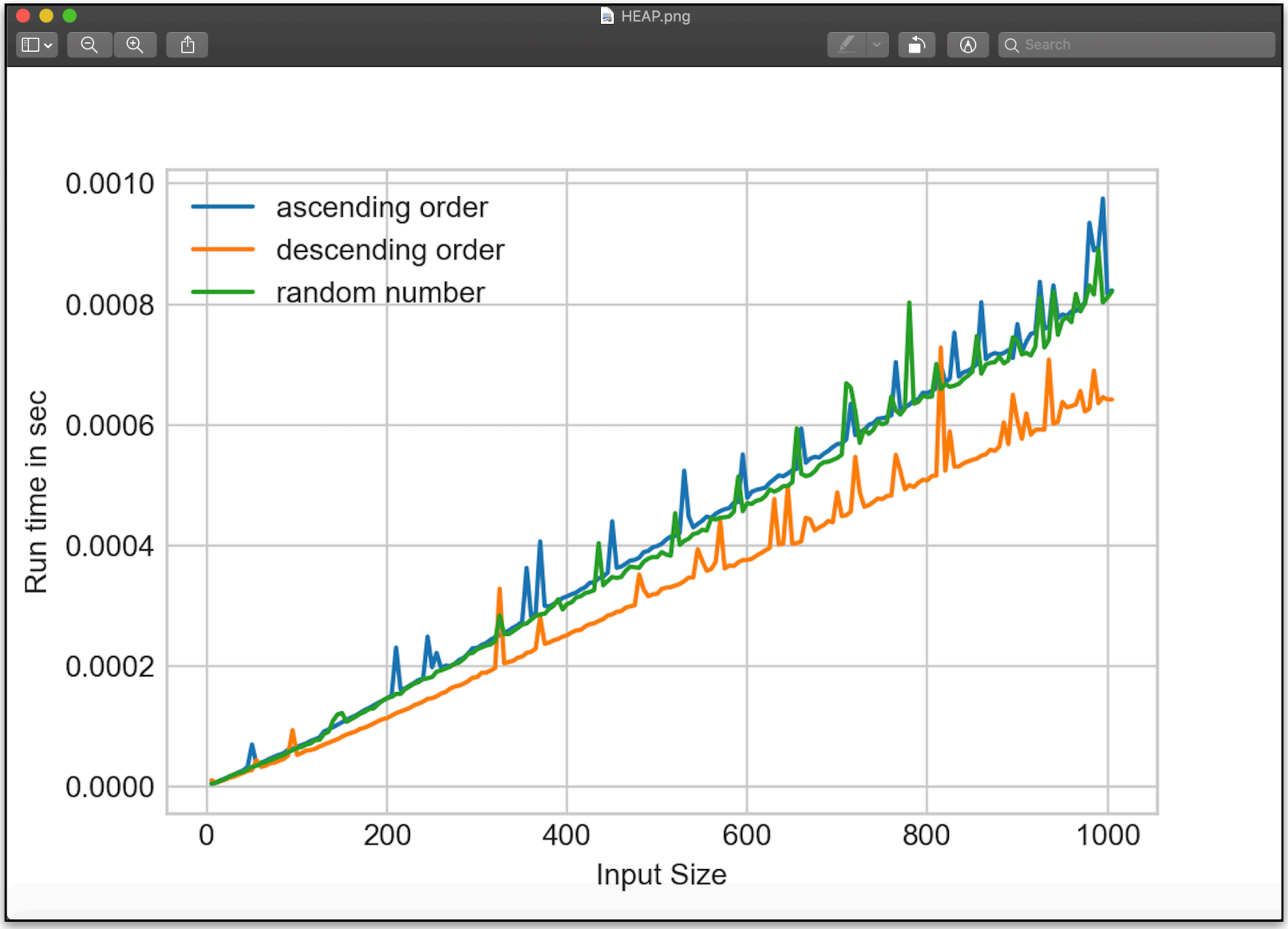


HEAP SORT :

- Heap sort is a comparison-based sorting algorithm.
- It divides its input into a sorted and an unsorted region, and it iteratively shrinks the unsorted region by extracting the largest element and moving that to the sorted region.
- Heap sort is slower than quick sort.
- Heap sort build a max heap from the input data.
- The largest item is stored at the root of the heap.
- It replace root with the last item of the heap followed by reducing the size of heap by 1. Finally, heapify the root of tree.
- BEST CASE OF HEAP SORT IS O(n log n) for distinct keys. And O(n) for equal key

- WORST CASE OF HEAP SORT IS O(n log n)

RUN TIME FOR DIFFERENT INPUT SIZE IS PLOTTED BELOW FOR HEAP SORT



QUICK SORT:

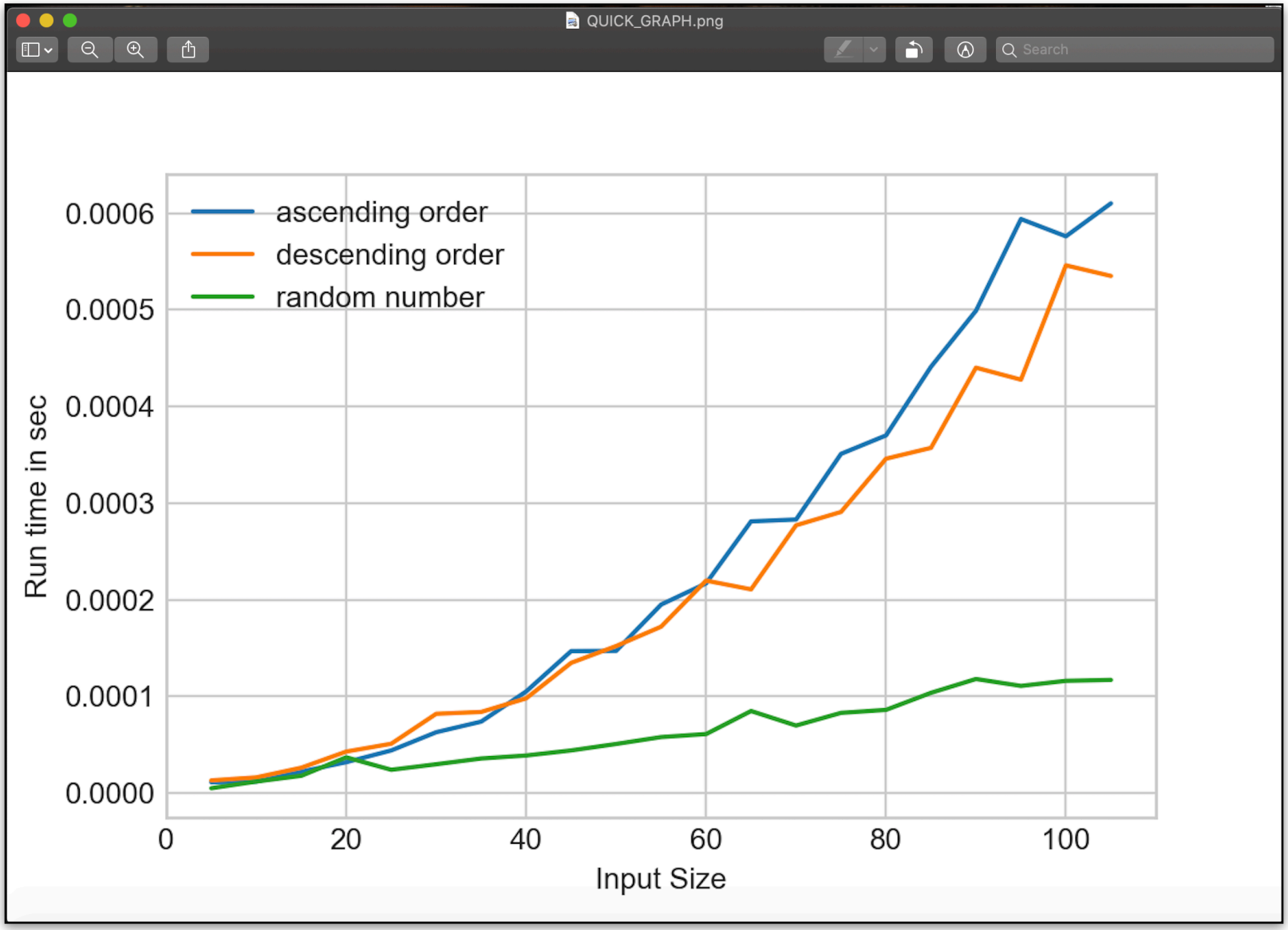
- QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.
- Efficient implementations of Quicksort are not a stable sort, meaning that the relative order of equal sort items is not preserved.
- The steps for median of three is :
    1. Pick an element, called a pivot, from the list.
    2. Reorder the list so that all elements which are less than the pivot come before the pivot and so that all elements greater than the pivot come after it (equal values can go either way).
    3. After this partitioning, the pivot is in its final position. This is called the partition operation.
    4. Recursively sort the sub-list of lesser elements and the sub-list of greater elements.
    - Quicksort with median-of-three partitioning functions nearly the same as normal quicksort with the only difference being how the pivot item is selected.
    - In normal quicksort the first element is automatically the pivot item. This causes normal quicksort to function very inefficiently when presented with an already sorted list.
    - The divison will always end up producing one sub-array with no elements and one with all the elements (minus of course the pivot item).
    - In quicksort with median-of-three partitioning the pivot item is selected as the median between the first element, the last element, and the middle element (decided using integer division of n/2).

- In the cases of already sorted lists this should take the middle element as the pivot thereby reducing the inefficency found in normal quicksort.

- BEST CASE OF QUICK SORT IS O(n)

- WORST CASE OF QUICK SORT IS O(n^2)

RUN TIME FOR DIFFERENT INPUT SIZE IS PLOTTED BELOW FOR QUICK SORT



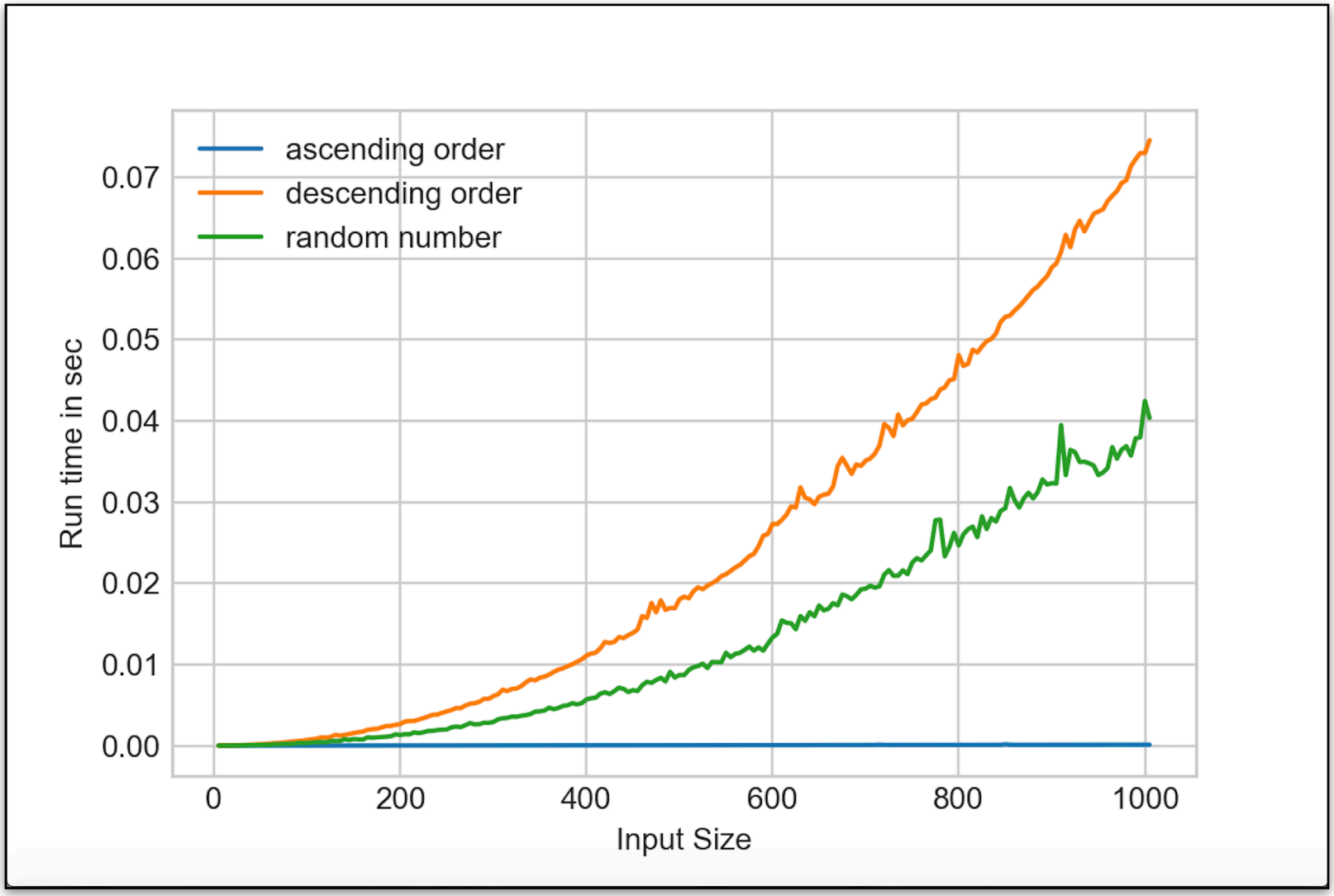
Insertion sort:

- Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time.
- Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list.
- At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.
- It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.

- **BEST CASE OF INSERTION SORT IS O(n)**

- **WORST CASE OF INSERTION SORT IS O(n ^ 2)**

RUN TIME FOR DIFFERENT INPUT SIZE IS PLOTTED BELOW FOR INSERTION SORT



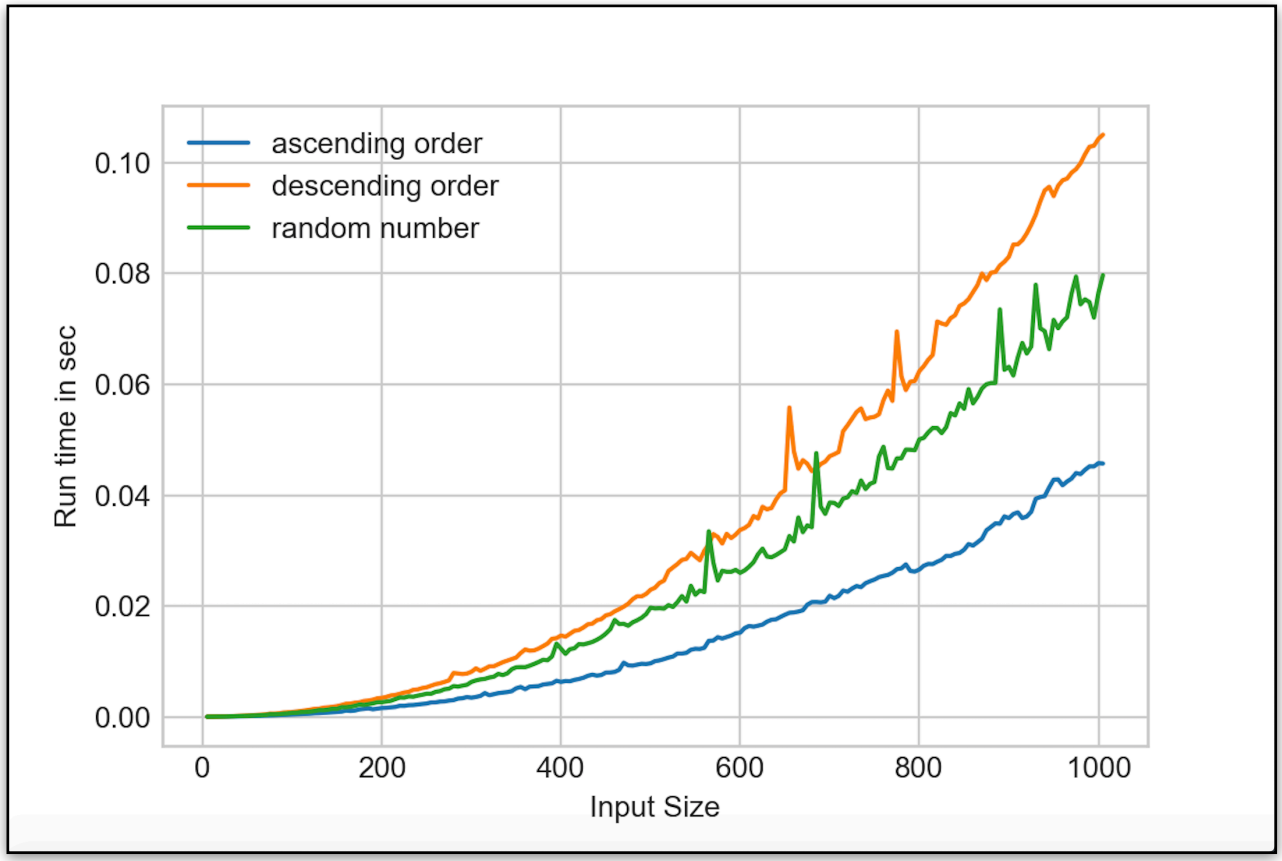
BUBBLE SORT:

- Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

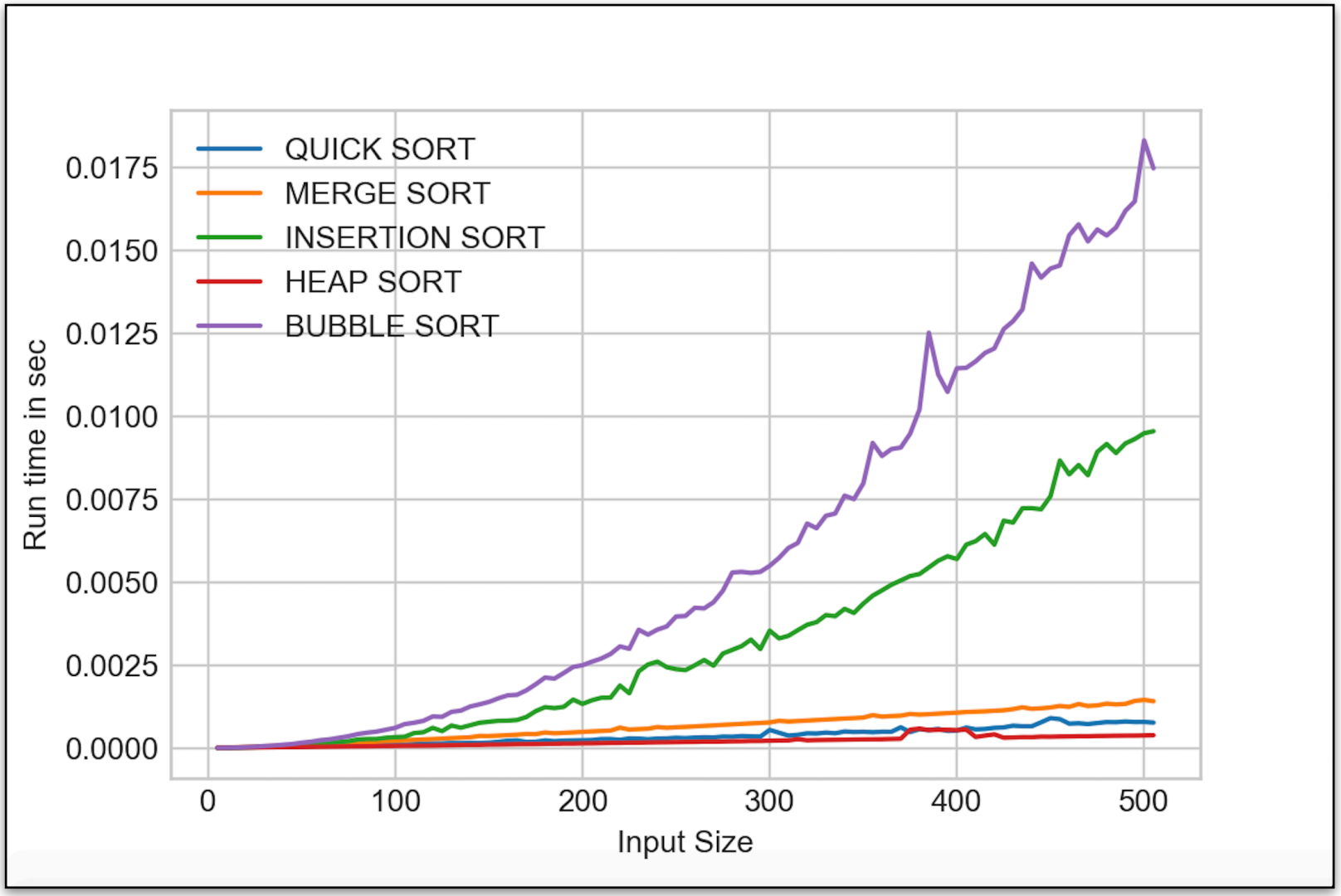
- **BEST CASE OF BUBBLE SORT IS_WHEN ELEMENTS IS ALREADY SORTED.**
- **Best case is O(n)**

- **WORST CASE OF BUBBLE SORT IS_WHEN ELEMENTS IS IN DESCENDING ORDER.**
- **Worst Case is O(n^2)**

RUN TIME FOR DIFFERENT INPUT SIZE IS PLOTTED BELOW FOR BUBBLE SORT



GRAPH COMPARING DIFFERENT RUN TIME FOR DIFFERENT INPUT SIZE USING ALL 5 ALGORITHM



QUICK SORT IS BETTER FOR LARGE DATA INPUTS