

CENTRAL UNIVERSITY OF HARYANA

Department of Computer Science & Engineering under SOET



ANALYSIS AND DESIGN OF ALGORITHMS

WAP to implement Bubble sort, Insertion Sort and Merge sort

Submitted by

Ajit Singh

Roll No- 192219

Semester- 5TH

Submitted to

Dr.Manish Kumar

Associate Professor

Central University of Haryana (SOET)

Assignment-2: Implement using Recursion method and plot the runtime for 100, 500, 1000, 1500, 2000, 2500, 3000 random integers.

BUBBLE SORT: Bubble sort arrange N numbers of array elements by placing the biggest element on proper position. It always arrange the data in descending order.

CODE:

```
def bubble_sort(alist):
    for i in range(len(alist) - 1, 0, -1):
        no_swap = True
        for j in range(0, i):
            if alist[j + 1] < alist[j]:
                alist[j], alist[j + 1] = alist[j + 1], alist[j]
                no_swap = False
        if no_swap:
            return

alist = input('Enter the list of numbers: ').split()
alist = [int(x) for x in alist]
bubble_sort(alist)
print('Sorted list: ', end='')
print(alist)
```

INSERTION SORT: Insertion sort arrange N numbers of elements of array by inserting particular item in a particular place such a way that the item are in sorted order.

CODE:

```
def insertion_sort(alist):
    for i in range(1, len(alist)):
        temp = alist[i]
        j = i - 1
        while (j >= 0 and temp < alist[j]):
            alist[j + 1] = alist[j]
            j = j - 1
        alist[j + 1] = temp

alist = input('Enter the list of numbers: ').split()
alist = [int(x) for x in alist]
insertion_sort(alist)
print('Sorted list: ', end='')
print(alist)
```

MERGE SORT: Merge sort is a sort algorithm that splits the items of array to be sorted order into two groups recursively sort each group and merge then into a final sorted sequence.

CODE:

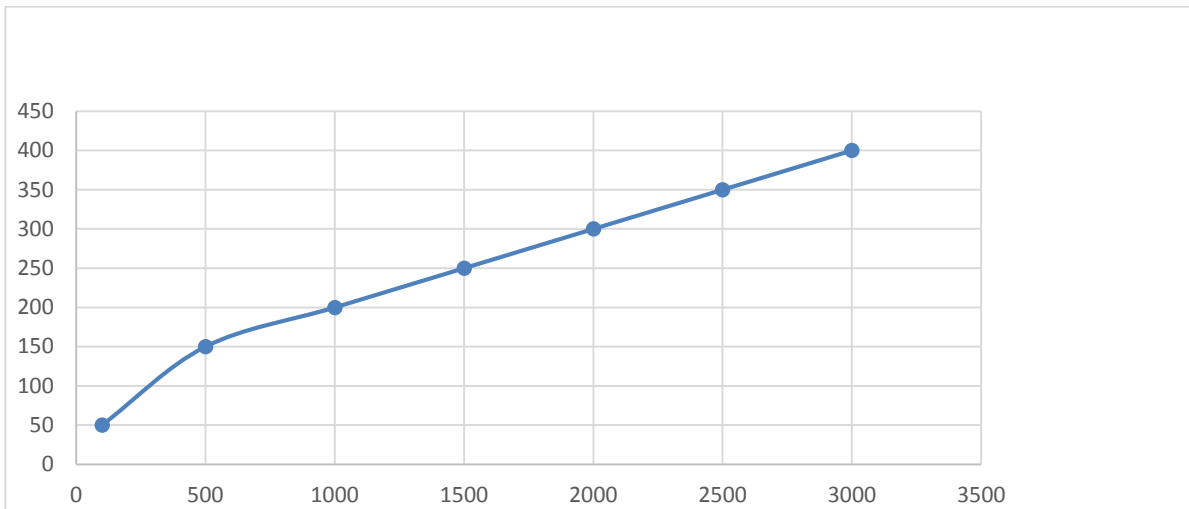
```
def merge_sort(alist, start, end):
    '''Sorts the list from indexes start to end - 1 inclusive.'''
    if end - start > 1:
        mid = (start + end)//2
        merge_sort(alist, start, mid)
        merge_sort(alist, mid, end)
        merge_list(alist, start, mid, end)

def merge_list(alist, start, mid, end):
    left = alist[start:mid]
    right = alist[mid:end]
    k = start
    i = 0
    j = 0
    while (start + i < mid and mid + j < end):
        if (left[i] <= right[j]):
            alist[k] = left[i]
            i = i + 1
        else:
            alist[k] = right[j]
            j = j + 1
        k = k + 1
    if start + i < mid:
        while k < end:
            alist[k] = left[i]
            i = i + 1
            k = k + 1
    else:
        while k < end:
            alist[k] = right[j]
            j = j + 1
            k = k + 1

alist = input('Enter the list of numbers: ').split()
alist = [int(x) for x in alist]
merge_sort(alist, 0, len(alist))
print('Sorted list: ', end='')
print(alist)
```

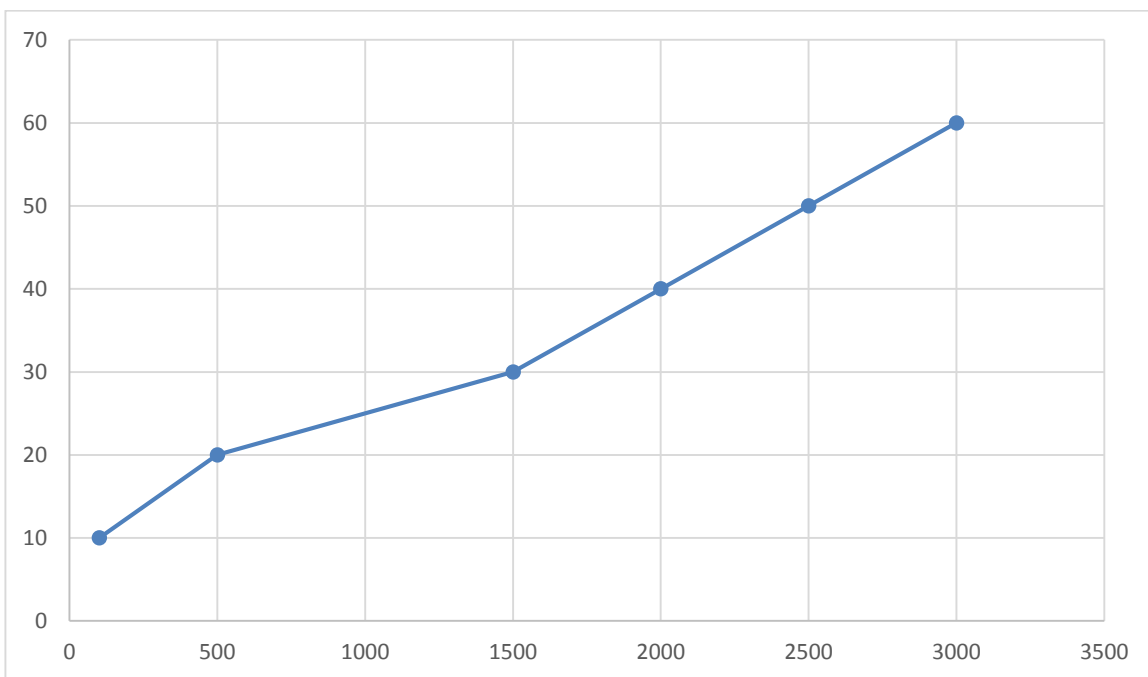
BUBBLE SORT:

GRAPH:



INSERTION SORT:

GRAPH:



MERGE SORT:

GRAPH:

