

Topics: To Compute the performance of sorting algorithms

INTRODUCTION

In this lab we would be evaluating the performance of sorting algorithm. Performance of bubble Sort, insertion sort, selection sort, merge sort & quick sort would be carried out.

EXERCISE

1. Implement insertion sort, selection sort bubble sort and merge sort. The number of inputs elements has to be passed from command line arguments. The elements has to be generated randomly within the code. Compute:
 - a. Check the performance of program by varying the number of elements.
 - b. Compute the time taken by each case (for particular number of inputs) $n = 10, 100, 1000, 10000, 1000000$.
 - c. plot a graph with number of inputs Vs time taken in seconds.
 - d. Compare the graphical plots for each sorting algorithms with its theoretical time complexity.
 - e. Also compute the time taken for sorted array (worst case) and compare with different number of elements. A sample example of sorting has been given below.

HELP

Taking input from command line arguments

There are two common ways to take arguments from command line and they are

- `sys.argv`
- `argparse` module

```
1 import sys
2 # total arguments
3 n = int(sys.argv[1])
4 print("The number of elements is :", n)
```

Run this at terminal -> python3 test.py 10

Random Numbers

The code to generate random numbers is given below

```
1 import random
2 # Generate 5 random numbers between 10 and 30
3 randomlist = random.sample(range(10, 30), 5)
4 print(randomlist)
```

Time

To compute the time taken by the program is given below

```
1 import time
2
3 # starting time
4 start = time.time()
5
6 # program body starts
7 for i in range(10):
8     print(i)
9
10 # sleeping for 1 sec to get 10 sec runtime
11 time.sleep(1)
12
13 # program body ends
14 # end time
15 end = time.time()
16
17 # total time taken
18 print(f"Runtime of the program is {end - start}")
```

EXAMPLE

A sample example to write sorting algorithm.

```
1 import time
2
3 def Sort_Algorithm(arr):
4     n = len(arr)
5     # Your Sorting Algorithm
6
7 def printList(arr):
8     for i in range(len(arr)):
9         print(arr[i], end = " ")
10    print()
11
12 # driver code to test the above code
13 if __name__ == '__main__':
14     arr =
15     print ("Given array is", end = "\n")
16     printList(arr)
17     # starting time
18     start = time.time()
19     # program body starts
20     Sort_Algorithm(arr) # put different Sorting Algorithms
21     # program body ends
22     # end time
23     end = time.time()
24     print("Sorted array is: ", end = "\n")
25     printList(arr)
26     # total time taken
27     print("Runtime of the program is {end - start}")
```