|  |  |
| --- | --- |
|  | **Manav Rachna University** |
| **Lab Assignment 1** |
| **Subject:** Analysis and Design of Algorithms **Subject Code**:CSE 204 B-T/P  **Semester: III/V** | |

**Learning Objective:** Students would be Able to implement sorting algorithms

**Learning Outcome:** To learn an assortment of sorting algorithms; and, from these, that different algorithms have properties making them appropriate for different applications

1. Write a Program to Sort a given set of elements using quick sort method and determine the time required to sort the elements.

Code:

import time

def partition(arr, low, high):

    pivot = arr[high]

    i = low - 1

    for j in range(low, high):

        if arr[j] <= pivot:

            i += 1

            arr[i], arr[j] = arr[j], arr[i]

    arr[i + 1], arr[high] = arr[high], arr[i + 1]

    return i + 1

def quick\_sort(arr, low, high):

    if low < high:

        pi = partition(arr, low, high)

        quick\_sort(arr, low, pi - 1)

        quick\_sort(arr, pi + 1, high)

arr = [10, 7, 8, 9, 1, 5, 12, 3, 2, 6]

n = len(arr)

start\_time = time.time()

quick\_sort(arr, 0, n - 1)

end\_time = time.time()

time\_taken = (end\_time - start\_time)\*1000000

print("Sorted array:", arr)

print(f"Time taken to sort the elements: {time\_taken:.10f}")

Output:



1. Write a Program to Sort a given set of elements using bubble sort method and determine the time required to sort the elements

Code:

import time

def bubble\_sort(arr):

    n = len(arr)

    for i in range(n - 1):

        swapped = False

        for j in range(n - 1 - i):

            if arr[j] > arr[j + 1]:

                arr[j], arr[j + 1] = arr[j + 1], arr[j]

                swapped = True

        if not swapped:

            break

arr = [10, 7, 8, 9, 1, 5, 12, 3, 2, 6]

n = len(arr)

start\_time = time.time()

bubble\_sort(arr)

end\_time = time.time()

time\_taken = (end\_time - start\_time)\*1000000

print("Sorted array:", arr)

print(f"Time taken to sort the elements: {time\_taken:.10f} seconds")

Output:

