|  |  |
| --- | --- |
|  | DEPARTMENT OF ARTIFICIAL INTELLIGNECE & DATA SCIENCE |

|  |  |
| --- | --- |
| Subject: Analysis of Algorithm | Course Code: CSC402 |
| Semester: 4 | Course: AI & DS |
| Laboratory No: 205 | Name of Subject Teacher: Pramod Bhavarthe |
| Name of Student: Sahil Shaikh | Roll Id: VU2S2223012 |

|  |  |
| --- | --- |
| Title of Practical | Implement Selection sort. |

**Theory –**

**Selection sort** is a simple sorting algorithm that sorts an array by repeatedly finding the minimum element from the unsorted part of the array and placing it at the beginning of the array. The algorithm maintains two sub-arrays: one that is sorted and another that is unsorted.

Here are the steps to perform selection sort on an array:

* Set the first element of the array as the minimum element.
* Compare the minimum element with the second element. If the second element is smaller, set it as the new minimum.
* Continue comparing each element with the minimum element until the end of the array is reached.
* If the minimum element is not the first element of the array, swap it with the first element.
* Now the first element of the array is sorted, so repeat steps 1 to 4 for the remaining unsorted part of the array.

**Time Complexity:** The time complexity of Selection Sort is O(N2) as there are two nested loops:

* One loop to select an element of Array one by one = O(N)
* Another loop to compare that element with every other Array element = O(N)

Therefore, overall complexity = O(N) \* O(N) = O(N\*N) = O(N2)

**Is Selection Sort Algorithm stable?**

* [**Stability:**](https://www.geeksforgeeks.org/stability-in-sorting-algorithms/)The default implementation is **not stable**. However, it can be made stable.

**Is Selection Sort Algorithm in place?**

* Yes, it does not require extra space.

**Advantages of Selection Sort:**

**Simplicity:** Selection sort is one of the simplest sorting algorithms to implement, making it easy to understand and use.

**In-place Sorting:** Selection sort performs in-place sorting, meaning it does not require any extra memory to sort the elements.

**Memory Efficient:** Selection sort uses a minimal amount of memory to sort the elements, making it ideal for use in memory-constrained environments.

**Disadvantages of Selection Sort:**

**Time Complexity:** The time complexity of selection sort is O(n^2), which makes it inefficient for large arrays or data sets. This means that selection sort is not the best choice when efficiency is critical.

**Performance:** Selection sort performs poorly on partially sorted arrays, as it does not take into account the already sorted elements.

**Unstable Sorting:** Selection sort is an unstable sorting algorithm, which means that it does not maintain the relative order of equal elements.

Overall, selection sort is a simple and memory-efficient algorithm, but it is not suitable for large data sets or situations where performance is critical.

**Program –**

import sys

A = [64, 25, 12, 22, 11]

# Traverse through all array elements

for i in range(len(A)):

    # Find the minimum element in remaining

    # unsorted array

    min\_idx = i

    for j in range(i+1, len(A)):

        if A[min\_idx] > A[j]:

            min\_idx = j

    # Swap the found minimum element with

    # the first element

    A[i], A[min\_idx] = A[min\_idx], A[i]

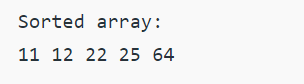
# Driver code to test above

print ("Sorted array")

for i in range(len(A)):

    print("%d" %A[i],end=" , ")

**Output –**

****

**Conclusion –**

* Selection sort is a simple and easy-to-understand sorting algorithm that works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the list and moving it to the sorted portion of the list.
* This process is repeated for the remaining unsorted portion of the list until the entire list is sorted.
* It has a time complexity of O(n^2) in the worst and average case which makes it less efficient for large data sets.
* Selection sort is a stable sorting algorithm.
* It can be used to sort different types of data.
* It has specific applications where it is useful such as small data sets and memory-constrained systems.

|  |  |  |  |
| --- | --- | --- | --- |
| **Grade and Dated Signature of Teacher** | **Total (10)** | **Remark** | **Dated signature of teacher** |
|  |  |  |