

3. Implement Greedy search algorithm for Selection Sort

Python Code:

```
def selection_sort(array):
    length = len(array)

    for i in range(length - 1):
        minIndex = i

        for j in range(i + 1, length):
            if array[j] < array[minIndex]:
                minIndex = j

        array[i], array[minIndex] = array[minIndex], array[i]

    return array
```

creating an empty list

```
lst = []
```

number of elements as input

```
n = int(input("Enter number of elements : "))
```

iterating till the range

```
for i in range(0, n):
```

```
    ele = int(input())
```

```
    # adding the element
```

```
    lst.append(ele)
```

```
print(lst)
```

```
print("The sorted array is: ", selection_sort(lst))
```

Java Code:

```
import java.io.*;
import java.lang.*;
import java.util.*;
public class Selectionsort {
    static void selectionSort(int[] A) {
        int[] U = A.clone();
        int n = A.length;
        for (int i = 0; i < n - 1; i++) {
            int min_idx = i;
            for (int j = i + 1; j < n; j++) {
                if (A[j] < A[min_idx]) {
                    min_idx = j;
                }
            }
            int tmp = A[i];
            A[i] = A[min_idx];
            A[min_idx] = tmp;
        }
    }
}
```

```

    }

    System.out.printf("Selection Sort:\nUnsorted array: %s\nSorted
array: %s", Arrays.toString(U), Arrays.toString(A));

}

public static void main(String ar[])
{

    System.out.println("Enter the size of array");
    Scanner sc=new Scanner(System.in);
    int n=sc.nextInt();
    int[] arr=new int[n];
    System.out.println("Enter Array elements");

    for(int i=0;i<n;i++)
    {
        arr[i]=sc.nextInt();
    }

    Selectionsort obj=new Selectionsort();
    obj.selectionSort(arr);
}

}

```