

Implement Greedy search algorithm for any of the following application: • Selection Sort • Minimum Spanning Tree • Single-Source Shortest Path Problem • Job Scheduling Problem • Prim's Minimal Spanning Tree Algorithm • Kruskal's Minimal Spanning Tree Algorithm • Dijkstra's Minimal Spanning Tree Algorithm.

def selection\_sort(arr):

n = len(arr)

# Traverse through all array elements

for i in range(n):

# Find the minimum element in the remaining unsorted array

min\_index = i

for j in range(i+1, n):

if arr[j] < arr[min\_index]:

min\_index = j

# Swap the found minimum element with the first element

arr[i], arr[min\_index] = arr[min\_index], arr[i]

print(f"Step {i+1}: {arr}") # Print the array after each swap

return arr

# Example usage:

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

print("Original array:", arr)

sorted\_arr = selection\_sort(arr)

print("Sorted array:", sorted\_arr)