Selection sort

def selectionSort(arr):

n = len(arr)

for i in range(n):

min\_idx = i

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

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

min\_idx = j

# Swap the found minimum element with the first element

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

return arr

# Main function

if \_\_name\_\_ == '\_\_main\_\_':

arr = list(map(int, input("Enter the elements separated by space: ").split()))

print("Original array:", arr)

sorted\_arr = selectionSort(arr)

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

theory

What is Selection Sort?

Selection Sort is a simple comparison-based sorting algorithm.

It repeatedly selects the minimum element from the unsorted portion and places it at the beginning.

🔹 How it Works:

Start from the first index and assume it's the minimum.

Compare it with all other elements to its right.

If a smaller element is found, update the minimum index.

After the inner loop, swap the minimum element with the current position.

Repeat this process for all elements.

🔹 Example:

For the input:

[64, 25, 12, 22, 11]

Step-by-step sorting:

Pass 1: Find min → 11, swap with 64 → [11, 25, 12, 22, 64]

Pass 2: Find min → 12, swap with 25 → [11, 12, 25, 22, 64]

Pass 3: Find min → 22, swap with 25 → [11, 12, 22, 25, 64]

Pass 4: Already sorted → [11, 12, 22, 25, 64]

🔹 Time Complexity:

Best case: O(n²)

Average case: O(n²)

Worst case: O(n²)

No matter what, it always runs in O(n²) time.

🔹 Space Complexity:

O(1) (in-place sorting, no extra memory used)