**//Assignment 6**

**Title:** Design and implement the Heap Sort algorithm to efficiently sort an array of integers in ascending order. The implementation should be optimized for time and space complexity and should clearly demonstrate the working principles of heap data structures (min-heap or max-heap as applicable)

```
def heapify(arr, n, i):
    """Ensures the subtree rooted at index i is a max-heap."""
    largest = i
    left = 2 * i + 1      # left child index
    right = 2 * i + 2     # right child index

    # Check if left child exists and is greater than root
    if left < n and arr[left] > arr[largest]:
        largest = left

    # Check if right child exists and is greater than current largest
    if right < n and arr[right] > arr[largest]:
        largest = right

    # If largest is not the root, swap and continue heapifying
    if largest != i:
        arr[i], arr[largest] = arr[largest], arr[i]
        heapify(arr, n, largest)
def heap_sort(arr):
    n = len(arr)
    # Build max-heap
    for i in range(n // 2 - 1, -1, -1):
        heapify(arr, n, i)
```

```
    print("\nMax Heap constructed:", arr)
    # Extract elements from heap one by one
    for i in range(n - 1, 0, -1):
        # Swap root (largest) with the last element
        arr[0], arr[i] = arr[i], arr[0]
        # Heapify the reduced heap
        heapify(arr, i, 0)
        print(f'Heap after removing element {n-i}:", arr)
def main():
    while True:
        print("\n--- Heap Sort Menu ---")
        print("1. Enter array")
        print("2. Perform Heap Sort")
        print("3. Exit")
        choice = input("Enter choice: ")
        if choice == "1":
            global arr
            arr = list(map(int, input("Enter integers separated by space: ").split()))
        elif choice == "2":
            if not arr:
                print("Array is empty! Please enter array first.")
            else:
                print("\nOriginal Array:", arr)
                heap_sort(arr)
                print("\nSorted Array (Ascending):", arr)
        elif choice == "3":
            print("Exiting program...")
```

```
            break
        else:
            print("Invalid choice! Please try again.")


if __name__ == "__main__":
    arr = []
    main()
```

//OUTPUT

--- Heap Sort Menu ---

1. Enter array

2. Perform Heap Sort

3. Exit

Enter choice: 1

Enter integers separated by space: 1 2 3 4 5 6 7 8

--- Heap Sort Menu ---

1. Enter array

2. Perform Heap Sort

3. Exit

Enter choice: 2

Original Array: [1, 2, 3, 4, 5, 6, 7, 8]

Max Heap constructed: [8, 5, 7, 4, 1, 6, 3, 2]

Heap after removing element 1: [7, 5, 6, 4, 1, 2, 3, 8]

Heap after removing element 2: [6, 5, 3, 4, 1, 2, 7, 8]

Heap after removing element 3: [5, 4, 3, 2, 1, 6, 7, 8]

Heap after removing element 4: [4, 2, 3, 1, 5, 6, 7, 8]

Heap after removing element 5: [3, 2, 1, 4, 5, 6, 7, 8]

Heap after removing element 6: [2, 1, 3, 4, 5, 6, 7, 8]

Heap after removing element 7: [1, 2, 3, 4, 5, 6, 7, 8]

Sorted Array (Ascending): [1, 2, 3, 4, 5, 6, 7, 8]

--- Heap Sort Menu ---

1. Enter array

2. Perform Heap Sort

3. Exit

Enter choice: