Assignment-59 : A Job Ready Bootcamp in C++, DSA and IOT

Queue

1. Create a Queue using Array or Dynamic Array.
2. Create a Queue using a linked list.
3. Efficiently implement k Queues in a single array.
4. Implement Priority Queue using Linked Lists.
5. Given Nodes with their priority, implement a priority queue using a doubly linked list.
6. Implementation of Deque using doubly linked list
   (7,8,9 You can also use STL for this problems)
7. Give an algorithm for reversing a queue Q. Only the following standard operations are allowed on queue.
   a. **enqueue(x):** Add an item x to the rear of the queue.
   b. **dequeue():** Remove an item from the front of the queue.
   c. **empty():** Checks if a queue is empty or not.
8. Given an integer k and a queue of integers, The task is to reverse the order of the first k elements of the queue, leaving the other elements in the same relative order. Only following standard operations are allowed on queue.
   a. **enqueue(x) :** Add an item x to rear of queue
   b. **dequeue() :** Remove an item from front of queue
   c. **size() :** Returns the number of elements in the queue.
   d. **front() :** Finds front item.
9. Given a queue with random elements, we need to sort it. We are not allowed to use extra space. The operations allowed on queue are :
   a. **enqueue(x) :** Add an item x to rear of queue
   b. **dequeue() :** Remove an item from front of queue
   c. **isEmpty() :** check if a queue is empty.
10. There are given N ropes of different lengths, we need to connect these ropes into one rope. The cost to connect two ropes is equal to the sum of their lengths. The task is to connect the ropes with minimum cost. Given N size array arr[] contains the lengths of the ropes.

    Example 1:

    Input:
    n = 4
    arr[] = {4, 3, 2, 6}
    Output:
    29
    Explanation:
    We can connect the ropes in following ways.
    1) First connect ropes of lengths 2 and 3. Which makes the array {4, 5, 6}. Cost of
    this operation 2+3 = 5.
    2) Now connect ropes of lengths 4 and 5. Which makes the array {9, 6}. Cost of
    this operation 4+5 = 9.

3) Finally connect the two ropes and all ropes have connected. Cost of this operation 9+6 =15
Total cost for connecting all ropes is 5 + 9 + 15 = 29. This is the optimized cost for connecting ropes. Other ways of connecting ropes would always have same or more cost. For example, if we connect 4 and 6 first (we get three rope of 3, 2 and 10), then connect 10 and 3 (we get two rope of 13 and 2). Finally we connect 13 and 2. Total cost in this way is 10 + 13 + 15 = 38.

Example 2:
Input:
n = 5
arr[] = {4, 2, 7, 6, 9}
Output:
62
Explanation:
First, connect ropes 4 and 2, which makes the array {6,7,6,9}. Cost of this operation 4+2 = 6. Next, add ropes 6 and 6, which results in {12,7,9}. Cost of this operation 6+6 = 12. Then, add 7 and 9, which makes the array {12,16}. Cost of this operation 7+9 = 16. And finally, add these two which gives {28}. Hence, the total cost is 6 + 12 + 16 +
28 = 62.