

Intuition:

1. To Intuition is to maintain two priority queues (pq1 and pq2) that store the candidates' smallest costs from the beginning and end of the list.
2. In each iteration, the code compares the smallest costs from pq1 and pq2 and selects the one with the lowest value. The corresponding cost is added to the total cost (ans), and the element is removed from the respective priority queue.
3. This process continues for k iterations, and at the end, the accumulated ans value represents the minimum total cost required to hire k workers.

Explanation:

1. Two integer variables i and j are initialized to track the start and end indices of the costs vector, respectively. These indices will be used to iterate over the vector.
2. Two priority queues, pq1 and pq2, are created. Priority queues are data structures that allow efficient insertion and retrieval of elements while maintaining a specific order. In this case, the elements are integers, and the order is ascending (greater<int>).
3. A long long variable ans is initialized to store the total cost.
4. The code enters a loop that executes k times, where k is the number of workers to be hired.
5. Inside the loop, two nested while loops are used to populate pq1 and pq2 with candidates from the costs vector. The conditions for these loops are as follows:
 - pq1 should contain at most candidates elements, and i should be less than or equal to j.
 - pq2 should also contain at most candidates elements, and i should be less than or equal to j.These loops essentially fill the priority queues with the smallest candidates elements from the costs vector, taking elements from both ends of the vector (i and j).
6. After the while loops, the code compares the top elements (t1 and t2) of pq1 and pq2 to determine which one is smaller. If both queues are empty, INT_MAX (maximum value for an int) is used as a placeholder.
7. If t1 is smaller than or equal to t2, the cost of the worker represented by t1 is added to ans, and the top element is removed from pq1 using pq1.pop(). Otherwise, the cost of the worker represented by t2 is added to ans, and the top element is removed from pq2 using pq2.pop().
8. The loop continues until k workers have been hired.
9. Finally, the function returns the accumulated total cost (ans).