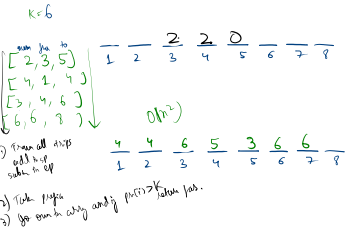
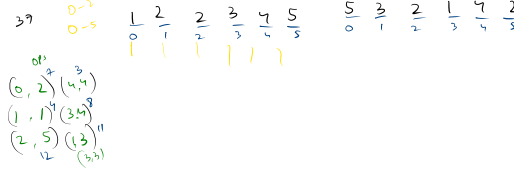


There is a car with 3 empty seats. The vehicle only drives east (i.e., it cannot turn around and drive west).
You are given the integer capacity and an array trips where trips[i] = (start, end, passengers) represents the number of passengers to pick up or drop off at each station. The stations are given as an array of integers, and the stations to pick up and drop off are there and not respectively. The locations are given as the number of kilometers from the start to the next station.
Return true if it is possible to pick up and drop off all passengers for all the given trips, or false otherwise.



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

pref[0] = pref[0]
pref[1] = pref[1]

0 4
if (l == 0) return pref[0];
return pref[0] - pref[l-1]

0(n^2)
0(a)

pref[i] = arr[i] + pref[i-1]

```
public int answerQueries(int[] pref, int l, int r) {  
    if(l==0) return pref[r];  
    return pref[r] - pref[l-1];  
}  
  
public int[] sumQuery(int[] arr, int[][] queries) {  
    // Your code here.  
    int n = arr.length;  
    //Step 1: create prefix sum array  
    int[] pref = new int[n];  
    pref[0] = arr[0];  
    for(int i = 1; i < n; i++){  
        pref[i] = arr[i] + pref[i-1];  
    }  
  
    int[] res = new int[queries.length];  
    for(int i = 0; i < queries.length; i++){  
        int l = queries[i][0];  
        int r = queries[i][1];  
        res[i] = answerQuery(pref, l, r);  
    }  
    return res;  
}
```

