

27) You are given an array `nums` consisting of integers. You are also given a 2D array `queries`, where `queries[i] = [posi, xi]`. For query `i`, we first set `nums[posi]` equal to `xi`, then we calculate the answer to query `i` which is the maximum sum of a subsequence of `nums` where no two adjacent elements are selected. Return the sum of the answers to all queries. Since the final answer may be very large, return it modulo $10^9 + 7$. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

CODE:

```
def maxSumAfterQueries(nums, queries):
    mod = 10**9 + 7
    total_sum = sum(x for x in nums if x > 0)
    result = []

    for pos, val in queries:
        prev_val = nums[pos]
        nums[pos] += val
        if prev_val > 0:
            total_sum -= prev_val

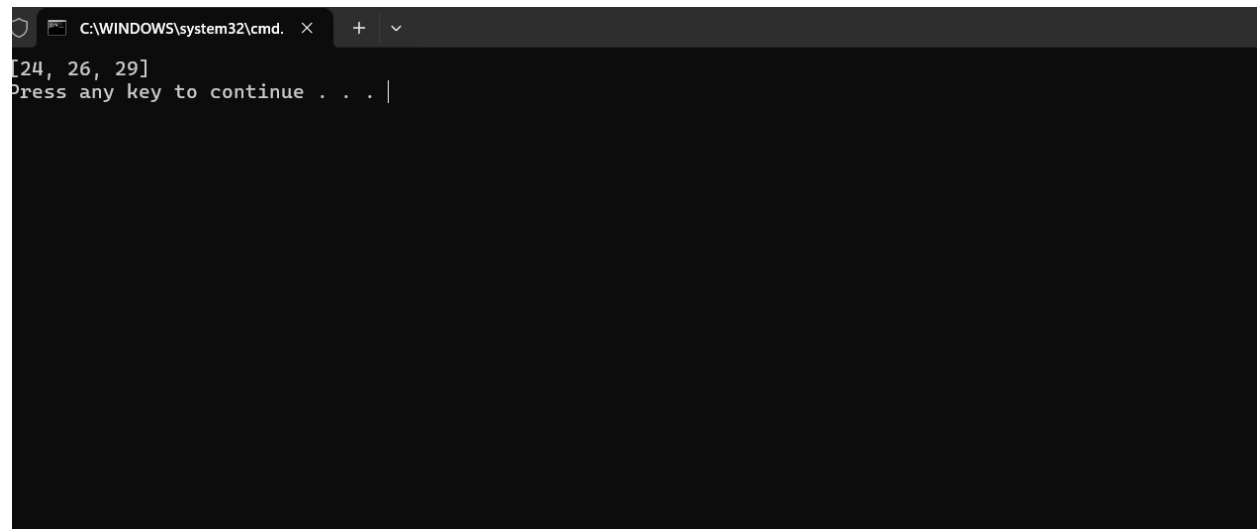
        if nums[pos] > 0:
            total_sum += nums[pos]

        result.append(total_sum % mod)

    return result

nums = [3, 6, 2, 8]
queries = [[0, 5], [1, 2], [2, 3]]
print(maxSumAfterQueries(nums, queries))
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the output of the Python code: '[24, 26, 29]' followed by a prompt 'Press any key to continue . . . |'.

TIME COMPLEXITY : $O(n+q)$