

28. 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:

```
MOD = 10**9 + 7
def max_sum_no_adjacent(nums):
    if not nums:
        return 0
    n = len(nums)
    if n == 1:
        return nums[0]

    include = nums[0]
    exclude = 0
    for i in range(1, n):
        new_include = exclude + nums[i]
        new_exclude = max(include, exclude)

        include = new_include
        exclude = new_exclude

    return max(include, exclude)

def process_queries(nums, queries):
    total_sum = 0

    for pos, val in queries:
        nums[pos] = val
        max_sum = max_sum_no_adjacent(nums)
        total_sum = (total_sum + max_sum) % MOD

    return total_sum
```

```
nums = [1, 2, 3, 4]
queries = [[0, 2], [3, 1], [1, 3]]

result = process_queries(nums, queries)
print(result)
```

Output:

```
16
```

Time Complexity:

- $T(n) = O(n)$