Q).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 109 + 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.

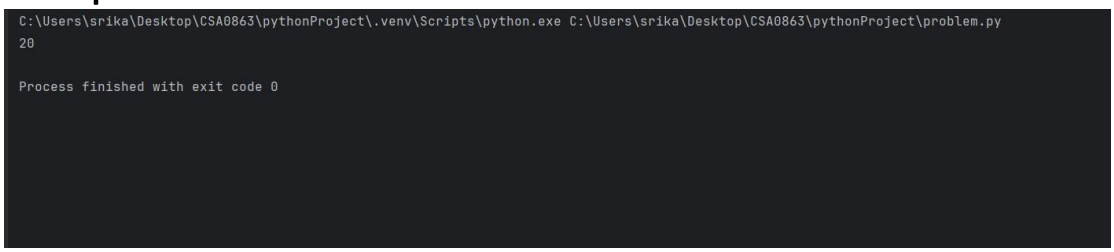
Program:

```
MOD = 10**9 + 7
def max_sum_no_adjacent(nums):
    n = len(nums)
    if n == 0:
        return 0
    if n == 1:
        return nums[0]
    dp_incl = [0] * n
    dp_excl = [0] * n
    dp_incl[0] = nums[0]
    dp_excl[0] = 0
    for i in range(1, n):
```

```
        dp_incl[i] = dp_excl[i-1] + nums[i]
        dp_excl[i] = max(dp_incl[i-1], dp_excl[i-1])
    return max(dp_incl[-1], dp_excl[-1])

def solve(nums, queries):
    total_sum = 0
    for posi, xi in queries:
        nums[posi] = xi
        total_sum += max_sum_no_adjacent(nums)
        total_sum %= MOD
    return total_sum
nums = [1, 2, 3, 4]
queries = [[0, 2], [1, 3], [2, 4]]
print(solve(nums, queries))
```

Output:



Time complexity:O(q*n)