

8. 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.

$\text{MOD} = 10^9 + 7$

```
def max_sum_no_adjacent(nums):
    incl, excl = 0, 0
    for num in nums:
        new_excl = max(incl, excl)
        incl = excl + num
        excl = new_excl
    return max(incl, excl)

def solve(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

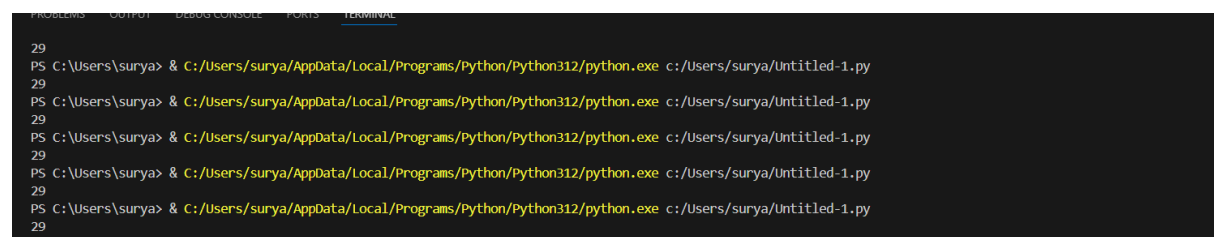
nums = [1, 2, 3, 4]
queries = [[1, 5], [0, 2], [3, 6]]
print(solve(nums, queries))

INPUT:[,2,3,4],[1,5],[0,2],3,6]
```

TIME COMPLEXITY:

$O(q \cdot n)$

Output:



```
29
PS C:\Users\surya> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Untitled-1.py
29
PS C:\Users\surya> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Untitled-1.py
29
PS C:\Users\surya> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Untitled-1.py
29
PS C:\Users\surya> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Untitled-1.py
29
PS C:\Users\surya> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Untitled-1.py
29
PS C:\Users\surya>
```

