

#!/bin/python3

import math

import os

import random

import re

import sys

#

# Complete the 'getMax' function below.

#

# The function is expected to return an INTEGER\_ARRAY.

# The function accepts STRING\_ARRAY operations as parameter.

#

def getMax(operations):

    stack = list()

    sol = list()

    max\_number = [0]

    for i in operations:

        if i[0]=='1':

            number = int(i[2:])

            stack.append(number)

            if number >= max\_number[-1]:

                max\_number.append(number)

        elif i[0]=='2' and stack:

            poppednumber = stack.pop()

            if poppednumber==max\_number[-1]:

                max\_number.pop()

        elif i[0]=='3':

            sol.append(max\_number[-1])

    return sol

    # Write your code here

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    n = int(input().strip())

    ops = []

    for \_ in range(n):

        ops\_item = input()

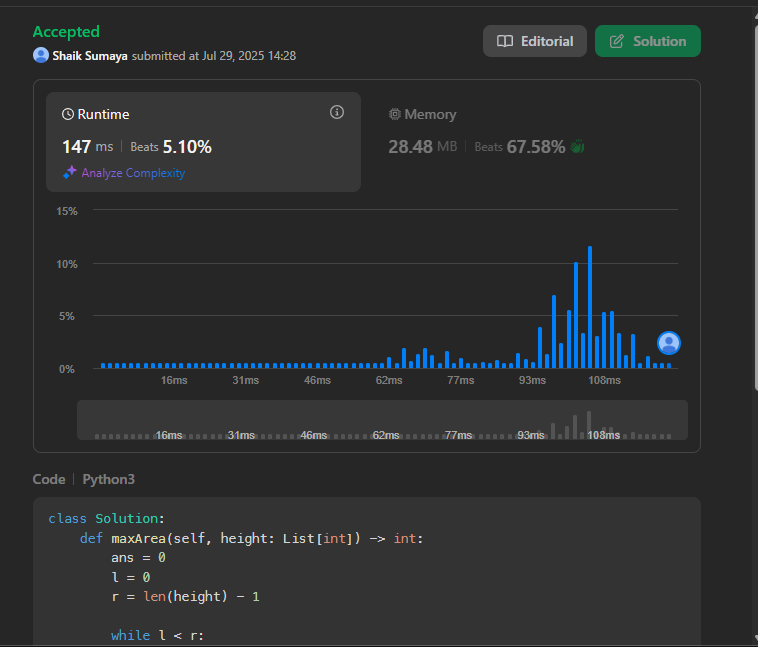
        ops.append(ops\_item)

    res = getMax(ops)

    fptr.write('\n'.join(map(str, res)))

    fptr.write('\n')

    fptr.close()



class Solution:

    def maxArea(self, height: List[int]) -> int:

        ans = 0

        l = 0

        r = len(height) - 1

        while l < r:

            minHeight = min(height[l], height[r])

            ans = max(ans, minHeight \* (r - l))

            if height[l] < height[r]:

                l += 1

            else:

                r -= 1

        return ans



# cook your dish here

n = int(input())

a = list(map(int, input().split()))

max\_len = 0

start = 0

for i in range(n):

if a[i] == 0:

max\_len = max(max\_len, i - start)

start = i + 1

max\_len = max(max\_len, n - start)

print(max\_len)