# 復習済み

A-E

# A 11/22 String

正解

# B 1122 String

正解

# C 11/22 Substring

正解

# D 1122 Substring

正解

本番とほぼ同じようなアルゴリズム。本番はr, lでしゃくとり法だが、復習はiだけを見ていってる。

## 正解コード(復習)

N = int(input())

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

ans = 0

count = 0

i = 0

s = set()

while i < N:

if i+1 < N and A[i] == A[i+1]:

if A[i] in s:

ans = max(ans, count)

l = i-count

for j in range(l, i, 2):

if A[j] == A[i]:

break

else:

s.remove(A[j])

count -= 2

i += 2

else:

count += 2

s.add(A[i])

i += 2

else:

ans = max(ans, count)

s = set()

if i>0 and A[i] == A[i-1]:

s.add(A[i])

count = 2

i += 1

else:

count = 0

i += 1

ans = max(ans, count)

print(ans)

# E 11/22 Subsequence

解いてない

解説コードをそのまま。

## 正解コード

import bisect

def judge(m, l, r): #lからrの部分で2\*m+1の文字列は可能か

if m == 0:

id = bisect.bisect\_left(slash\_id, l)

if id != len(slash\_id) and slash\_id[id] <= r:

return True

else:

return False

i = bisect.bisect\_left(one\_id, l)

if i+m-1 >= len(one\_id):

return False

j = bisect.bisect\_left(slash\_id, one\_id[i+m-1])

if j == len(slash\_id):

return False

k = bisect.bisect\_left(two\_id, slash\_id[j])

if k+m-1 >= len(two\_id):

return False

if two\_id[k+m-1] <= r:

return True

else:

return False

N, Q = map(int, input().split())

S = input()

one\_id = []

two\_id = []

slash\_id = []

for i in range(N):

if S[i] == "1":

one\_id.append(i)

elif S[i] == "2":

two\_id.append(i)

else:

slash\_id.append(i)

for i in range(Q):

L, R = map(int, input().split())

L -= 1

R -= 1

ok = -1

ng = (R-L+2)//2

while ok+1 < ng:

mid = (ok+ng)//2

if judge(mid, L , R):

ok = mid

else:

ng = mid

if ok == -1:

print(0)

else:

print(2\*ok+1)

# F 1122 Subsequence

解いてない

# G Fibonacci Product

解いてない