# 復習済み

1. E

# A Hamming Distance

正解

# B Ranking with Ties

正解

ランキングは上位に何人いるかで求められる。

# C Make it Forest

正解

# D Switch Seats

正解

自分のコードでは、あるiに対してi-1とi+1を見ていたが、その必要はなくiとi+1を見ればいい。

## 正解コード

T = int(input())

for \_ in range(T):

N = int(input())

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

count = [0]\*N

matrix = [[set(), set()] for i in range(N)]

out = [False]\*N

ans = 0

for i in range(2\*N):

a = A[i]

if count[a-1] == 0:

count[a-1] = 1

for di in [-1, 1]:

if 0 <= i + di < 2\*N:

if A[i+di] > a:

matrix[a-1][0].add((A[i+di]-1, i+di))

elif A[i+di] == a:

out[a-1] = True

else:

for di in [-1, 1]:

if 0 <= i + di < 2\*N:

if A[i+di] > a:

matrix[a-1][1].add((A[i+di]-1, i+di))

elif A[i+di] == a:

out[a-1] = True

for i in range(N):

if out[i]:

continue

for num1, id1 in matrix[i][0]:

for num2, id2 in matrix[i][1]:

if num1 == num2 and id1 != id2 and out[num1] == False:

ans += 1

print(ans)

## 改善コード

T = int(input())

for \_ in range(T):

N = int(input())

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

id\_2 = [-1]\*(N+1)

ans = 0

for i in range(2\*N):

if id\_2[A[i]] == -1:

id\_2[A[i]] = 0

else:

id\_2[A[i]] = i

for i in range(2\*N-1):

a1 = A[i]

a2 = A[i+1]

a1\_id2 = id\_2[a1]

a2\_id2 = id\_2[a2]

if a1 == a2:

continue

if abs(i - a1\_id2) <= 1:

continue

if abs(i+1 - a2\_id2) <= 1:

continue

if abs(a1\_id2 - a2\_id2) <= 1:

ans += 1

print(ans)

# E Replace

不正解

基本解説通り。

1. S==T
2. Tの種類が26種類

の時はすぐに回答。

次にansに有向グラフの辺の数をセットする。

その後、連結成分ごとに考えるため、undirected\_graphで無向グラフにして、連結成分を抽出。連結成分ごとに見て、

1. 自己ループがあるなら ansから1引く
2. 連結成分の全てのノードでcycleを作っていたら ansに１足す
3. それ以外は何もしない

以上を経てansが答えとなる。

## 正解コード

from collections import defaultdict, deque

N = int(input())

S = input()

T = input()

if S == T:

print(0)

exit()

if len(set(T)) == 26:

print(-1)

exit()

adjacent = {}

for i in range(N):

if S[i] in adjacent:

if adjacent[S[i]] != T[i]:

print(-1)

exit()

else:

adjacent[S[i]] = T[i]

ans = len(adjacent.keys())

undirected\_graph = defaultdict(set)

visited = {}

nodes = set()

for k, v in adjacent.items():

undirected\_graph[k].add(v)

undirected\_graph[v].add(k)

visited[k] = False

visited[v] = False

nodes.add(k)

nodes.add(v)

connected\_components = []

for node in nodes:

if visited[node]:

continue

visited[node] = True

com = set([node])

q = deque([node])

while q:

now = q.popleft()

for nxt in undirected\_graph[now]:

if visited[nxt]:

continue

visited[nxt] = True

q.append(nxt)

com.add(nxt)

connected\_components.append(com)

for com in connected\_components:

self\_loop = False

for node in com:

if node not in adjacent:

continue

if node == adjacent[node]:

self\_loop = True

ans -= 1

break

if self\_loop:

continue

degree = {}

for node in com:

degree[node] = 0

for node in com:

if node not in adjacent:

continue

degree[adjacent[node]] += 1

all\_in\_cycle = True

for k, v in degree.items():

if v != 1:

all\_in\_cycle = False

break

if all\_in\_cycle:

ans += 1

print(ans)

# F Range Power Sum

解いてない

dpの方ではない解説と同じような考え。

## 正解コード

import math

def combi(n, k, MOD):

return (math.factorial(n) // (math.factorial(k)\*math.factorial(n-k))) % MOD

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

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

MOD = 998244353

Sk = [[1]\*(N+1) for \_ in range(K+1)]

Sk[1][0] = 0

ans = 0

for i in range(N):

Sk[1][i+1] = (Sk[1][i] + A[i]) % MOD

for k in range(2, K+1):

Sk[k][0] = 0

for i in range(1, N+1):

Sk[k][i] = (Sk[k-1][i] \* Sk[1][i]) % MOD

for k in range(K+1):

c = combi(K, k, MOD)

tmp\_s = 0

tmp\_s2 = 0

for i in range(1, N+1):

if k % 2 == 0:

tmp\_s = (tmp\_s + Sk[k][i-1]) % MOD

else:

tmp\_s = (tmp\_s + (-Sk[k][i-1]) % MOD ) % MOD

tmp\_s2 = (tmp\_s2 + Sk[K-k][i] \* tmp\_s) % MOD

ans = (ans + c\*tmp\_s2) % MOD

print(ans)

## 補足

combiの関数のところでのmodを計算するときを求めてからではなくモジュロ逆元を使うバージョンがある。それが以下である。しかしこの問題では時間がほぼ変わらなかった。

def modinv(x, mod):

return pow(x, mod - 2, mod)

def comb(n, k, mod):

if k > n or k < 0:

return 0

return (math.factorial(n) \* modinv(math.factorial(k), mod) % mod) \* modinv(math.factorial(n - k), mod) % mod

# G Colorful Spanning Tree

解いてない