Problem Solving 9

Tuesday, 7 September 2021 8:00 PM

Q1 You are given N elements. You have to make on the basis of elements (i) the difference (abs) b/w first and second element is minimum b/w second and third minimum

NON decreasing

$|a[i] - a[j]|$

$N > 10^5$

next_permutation

$N! \times N!$

If you want to generate all possible permutation

TC → $N \times N!$

$N! \times N! \times N$ TC

diff in non decreasing order

b/w second and last → should be minimum diff

$N!$

$(N \times N!) \times N$

reduce if difference than the original one

ans = [5, -2, 4, 8, 6, 5]

TC → N log N

Sorting

ans = [ ]

i = 0 j = n-1 k = n-1

while (i <= j)

ans[k--] = ans[j--]

ans[k--] = ans[i++]

max diff

next pos max diff

SC → O(N)

TC → O(N log N)

Q Given 2 sorted arrays. You have to find max sum path from starting at any of 2 arrays and end at any of the 2 arrays

→ You are allowed to move the other array if values at both position are equal

DISTINCT

max path sum =

16 10 15 17 18

3 5 7 14 15 16 17 18

INT MAX

INT MAX

ans[i] = ans[j] → if (sum1 > sum2) ans += sum1 else ans += sum2

sum1 = 0

sum2 = 0

ans[i] > ans[j] → j++ sum2 += ans[j]

ans[i] < ans[j] → j++ sum1 += ans[i]

5 min

ans = 6 + 7 = 13

INT_MAX

INT_MAX

sum1 = 0 sum2 = 0 ans = 0

i = 0 j = 0

while (i < N and j < M) {

if (ans[i] < ans[j]) {

sum1 += ans[i]

i++

}

else if (ans[i] > ans[j]) {

sum2 += ans[j]

j++

}

else {

ans += ans[i];

ans += max(sum1, sum2);

sum1 = sum2 = 0;

i++ j++

}

1 2 3 4 4 4 5

sum1 = 21 sum2 = ~~0~~ ~~10~~ 29

ans = 6 + 5 = 11

INT MAX

INT-MAX

while (i < n) sum1

- (j < m) sum2 -

ans += max(sum1, sum2)

Q Given an array of N integers (We can reorder the array)

There are Q queries each with 2 values L, R

You are asked to ans for each query sum of elements in subarray [L:R]

Find Optimal arrangement of the array so that sum of answers to all the above queries is maximum

Soln

Find the index that occured the most in all the queries.

and assign that index the max possible value

O(N+Q)

Hash map

→ for (q →)

for (i=L1 → R1)

mp[i] ++;

Q

L1 R1

L2 R2

L3 R3

Q × N

O(N+Q)

No. of times an index occured in all the queries is optimal time

count of how many times this index occured in all the queries.

for(i=0 → n)

→ sum += ans[i] * prefsum[i]

A = [5 3 2] 2, 5, 3

Q = (1 indexed) 3, 0, 2

[1, 2] → 8 7

[2, 3] → 5 8

[1, 3] → 10 10

ans[L] += 1

ans[R] -=

35

[[4] 5 [4 2]]

1,3 (1 indexed) → 0, 0

3,4 2,3

1,4

[0, 1)

N.A

< freq, index

10

i ≥ L

5 × 3 + 4 × 3 + 2 × 4 + 2 × 2

= 39

Q There are N mirrors and N mirrors.

Each mirror will be assigned to 1 mirror

Mirrors → placed on Y axis

Mirrors → placed on X axis

x location → y location

Energy used by mirror to mirror = $\sqrt{x^2+y^2}$

You have to assign mirror to mirror such that sum of energies of each mirror is minimum.

$\sqrt{a^2+b^2}$

$H = \sqrt{a^2+b^2}$

$\sqrt{x_1^2+y_1^2} + \sqrt{x_2^2+y_2^2} + \sqrt{x_3^2+y_3^2}$

mirrors = [ ] → sort

mirrors = [ ] → sort

$\sqrt{a[i]^2 + B[i]^2}$

$\sqrt{44+16} + \sqrt{9+25}$

$\sqrt{160} + \sqrt{34}$

18.48

To prove

AC + BD < AD + BC

In a triangle, sum of 2 sides is greater than the 3rd side

BO + CO > BC —(1)

AO + OD > AD —(2)

BO + CO + AO + OD > BC + AD

BD + AC > AD + BC

Q.E.D

Hence proved

Quite fairly done

$\sqrt{5^2+4^2} + \sqrt{12^2+5^2}$

for (int i)

ans += (val)

ans %= MOD

ans -= val

(ans + MOD) % MOD

Sort

(10 + 11 + 12) % 7

(7 + 3) % 7

3

INVERSE MODULO

10 % MOD

a/b % MOD

10^x pow(r, MOD-2) % MOD

a^x pow(b, MOD-2, MOD)

Binary Exponential

2^k

2 × 2 × ... × 2

2^k

$2^{k/2} \times 2^{k/2}$

log k