2 Pointers

Tuesday, 23 August 2021 1:30 PM

Intro:

Subpa Sol 2Ptr Grind

2 Years: Tracking Prog

Bad Break up with Array Pts (Early Fashion)

Have good Read writing skills (Hard)

Why can understand

This is an Intuitive Technique, no need to know any data structure to understand this.

Output is O(N)

Multiple Outputs will be what entire question, some info to annunciated at start, output will be strong

Contest Flow:

1) Intro to 2 pointer
2) a+b=K
3) a-b=k
4) a+b+c=k / a-b-c=k /
5) Min abs diff between 3 array
6) Why 2 Pointers work?
7) Why we make break with Array Things
8) Max water fetch

Note: We are all here to learn & have fun, if fun, & please don't spam, FB a humble request from my side.

Problems:

1) Given a sorted array, check if there exists a pair (i,j) such that arr[i]+arr[j] = K (K is Extra Space)

i	j
0	9
0	8
0	7
1	6
1	5
2	5
3	4

22 > 19, j--
15 > 19, j--
11 > 19, j--
8 < 19, i++
11 > 19, j--
9 < 19, i++
7 < 19, i++
11 > 19, j--
9 < 19, i++

2) Array not sorted?

→ (i,j) = k

arr[i] - arr[i] = k → -5, K
arr[j] - arr[j] = -k → 5, -K

2B) Given a sorted array, check if there exists a pair (i,j) such that (i≠j) & arr[i]-arr[j] = (K*5), K>0

Ex: arr[N] = {3, 0, 1, 3, 6, 8, 11, 14, 16, 25}

i	j	arr[i]-arr[j]
2	4	4-0 = 5
3	5	6-1 = 5
4	6	8-3 = 5

Intuitively 2 pointers

i	j
0	1
0	2
1	X 2
1	3
2	3

k=2

i	j	diff
0	1	3>2
0	2	4<5
0	3	6>5
1	3	3<5
1	4	6>5
2	4	5=5

3) Given a sorted array check if there exists a triplet (i,j,k) such that arr[i]+arr[j]+arr[k] = Sum & i≠j≠k

i	j	k	Sum
0	1	2	4<5
0	1	3	6>5
0	1	4	6>5
1	1	1	3<5
2	1	2	4<5
2	1	3	6>5
2	1	4	6>5
2	2	3	5=5

4) Given 3 sorted arrays A[], B[], C[], find (i,j,k) such that

max(A[i], B[j], C[k]) - min(A[i], B[j], C[k]) is minimum

Ex:

	0	1	2	3	4	5
A[]	4	6	20	23	40	50
B[]	23	29	30	35	50	
C[]	15	21	29	31	39	72

i	j	k	max(3,-4,-15)	min(3,-4,-15)	3-(-15)
0	0	0			
0	0	0			
5	5	5			
4	3	4			
4	3	3			

i	j	k	max	min	Diff
0	0	0	4	4	0
0	1	1	6	6	0
1	1	1	23	23	0
2	1	2	29	29	0
3	1	2	30	29	1
4	2	2	39	29	5
4	3	3	40	29	11
5	3	3	50	29	21
5	3	4	50	29	21

We make min pointer because our concern get any

Note: We make a pointer of the know that we cannot get any when that end.

T.C → O(N+M)

5) Given an array of integers, where each arr[i] represents height of each building. Find max amount of water between any 2 buildings

Note: Between any 2 chosen buildings water will never exceed height of 2 buildings.

Eg: arr[] = {3, 5, 4, 7, 3, 6, 2, 7, 1, 2}

i	j	area
0	1	5 × min(3,5) = 15
0	2	8 × min(3,4) = 16
0	3	21 × min(3,7) = 21
0	4	24 × min(3,3) = 24
0	5	27 × min(3,6) = 18
0	6	28 × min(3,2) = 6
0	7	35 × min(3,7) = 21
0	8	36 × min(3,1) = 3
0	9	38 × min(3,2) = 6

i	j	Area
0	1	15
0	2	16

6) Given a sorted array check if there exists a triplet (i,j,k) such that arr[i]+arr[j]+arr[k] = Sum & i≠j≠k

i	j	k	Sum	
0	1	2	4	4<5, Increase left
0	1	3	6	6>5, R-- decrease right
0	1	4	6	6>5, R-- decrease right
1	1	1	3	3<5, Increase left
2	1	2	4	4<5, R++
2	1	3	6	6>5, R-- decrease right
2	1	4	6	6>5, R-- decrease right
2	2	3	5	5=5 return True

- 1/2 Above technique can be applied a+b-c=k & a-b-c=k

7) Given an array of integers, where each arr[i] represents height of each building. Find max amount of water between any 2 buildings

Note: Between any 2 chosen buildings water will never exceed height of 2 buildings.

Eg: arr[] = {3, 5, 4, 7, 3, 6, 2, 7, 1, 2}

i	j	area
0	1	5 × min(3,5) = 15
0	2	8 × min(3,4) = 16
0	3	21 × min(3,7) = 21
0	4	24 × min(3,3) = 24
0	5	27 × min(3,6) = 18
0	6	28 × min(3,2) = 6
0	7	35 × min(3,7) = 21
0	8	36 × min(3,1) = 3
0	9	38 × min(3,2) = 6

i	j	Area
0	1	15
0	2	16

8) Given N array elements, where each arr[i] represents height of each building. Find max amount of water between any 2 buildings

Note: Between any 2 chosen buildings water will never exceed height of 2 buildings.

Eg: arr[] = {3, 5, 4, 7, 3, 6, 2, 7, 1, 2}

i	j	area
0	1	5 × min(3,5) = 15
0	2	8 × min(3,4) = 16
0	3	21 × min(3,7) = 21
0	4	24 × min(3,3) = 24
0	5	27 × min(3,6) = 18
0	6	28 × min(3,2) = 6
0	7	35 × min(3,7) = 21
0	8	36 × min(3,1) = 3
0	9	38 × min(3,2) = 6

i	j	Area
0	1	15
0	2	16

9) We can know how make a decision and either update P1 & P2

Ex: P1 & P2 are pointers where one gets an array or regular pointer

D Binary

1) Where to initialize:
2) How to update P1 & P2
3) While(condition)