Todays Content

1. Min Triplet pairs
2. Water Logging

18 Given 3 Sorted Arrays A[x] B[y] C[z]

Return min of expression { Max(A[i], B[j], C[k]) - Min(A[i], B[j], C[k]) }

Where i, j, k are indices.

	0	1	2	3
A[4] = {	3	14	16	23 }
B[4] = {	-6	23	24	30 }
C[4] = {	-15	15	26	31 }

i	j	k	Max(A[i], B[j], C[k]) - Min(A[i], B[j], C[k])
0	0	0	max(3, -6, -15) - min(3, -6, -15) = 3 - (-15) = 18
0	0	1	max(3, -6, 15) - min(3, -6, 15) = 15 - (-6) = 21
3	3	3	max(23, 30, 31) - min(23, 30, 31) = 31 - 23 = 8
3	2	2	max(23, 24, 26) - min(23, 24, 26) = 26 - 23 = 3

Idea: Generate all triplets

Calculate expression value & get overall min.

```
for(i=0; i<N; i++){                    TC: O(N3)  SC: O(1)
    for(j=0; j<N; j++){
        for(k=0; k<N; k++){
            ans = min(ans, max(A[i], B[j], C[k]) - min(A[i], B[j], C[k]));
        }
    }
}
return ans;
```

Idea2: Fix a pair for A[] & B[]

Try search for C[i] : greatest ele <= a or 1
smallest ele >= b or 2
smallest ele >= a or greatest <= b.

C [a c b] c
1 3 2

TC: O(N^2 * 3 log N)

Idea3: Fix every A[i] as a:

For every a:

Calculate flor & cil of a in B[] & C[].

Case1: [a b c # cil of a in both B[] & C[] = b, c

Expected = max(b, c) - a

Case2:

2a b a c # flor of a in B[] cil of a in C[]

2b c a b # flor of a in C[] cil of a in B[]

Case3 [b c a # flor of a in both B[] & C[] = b, c

Exp = a - min(b, c);

TC: O(N * 4 log N) = O(N log N)

Idea3:

	0	1	2	3
A[] = {	10	14	16	23 }
		↑		
B[] = {	4	23	24	30 }
		↑		
C[] = {	7	10	26	31 }
			↑	

{ 7 10 30 } = 23

Hard gesture pointer update

i	j	k	Max(A[i], B[j], C[k]) - Min(A[i], B[j], C[k])		Update
0	0	0	Max(10, 4, 7) - Min(10, 4, 7)	=	6 more small P_2
0	1	0	Max(10, 23, 7) - Min(10, 23, 7)	=	16
0	1	1	Max(10, 23, 10) - Min(10, 23, 10)	=	13
1	1	1	Max(14, 23, 10) - Min(14, 23, 10)	=	13
1	1	2	Max(14, 23, 26) - Min(14, 23, 26)	=	12
2	1	2	Max(16, 23, 26) - Min(16, 23, 26)	=	10
3	1	2	Max(23, 23, 26) - Min(23, 23, 26)	=	3

#Idea:

$P_1 = 0, P_2 = 0, P_3 = 0$

while($P_1 <$ A.size(), P_2 · B.size(), P_3 · size()){

    calculate up val & update ans

    move pointer with min val

}

Why ?

	0	1	2	3
A[] = {	10	14	16	23 }
B[] = {	4	23	24	30 }
C[] = {	7	10	26	31 }

P_1	P_2	P_3	Max(A[i], B[j], C[k]) - Min(A[i], B[j], C[k])
0	0	0	max(10, 4, 7) - min(10, 4, 7) # Move min pointer?

{ 4 7 10 }

Here: 4 with smallest ele in other array Diff: 4.

diff 4 with other ele in arr[] diff > 4, diff a guarantee that with 4, we cannot get better ans hence discard.

Tc: O(N+M+N);

```
int minDiff(vector<int> &A, vector<int> &B, vector<int> &C){

}
```

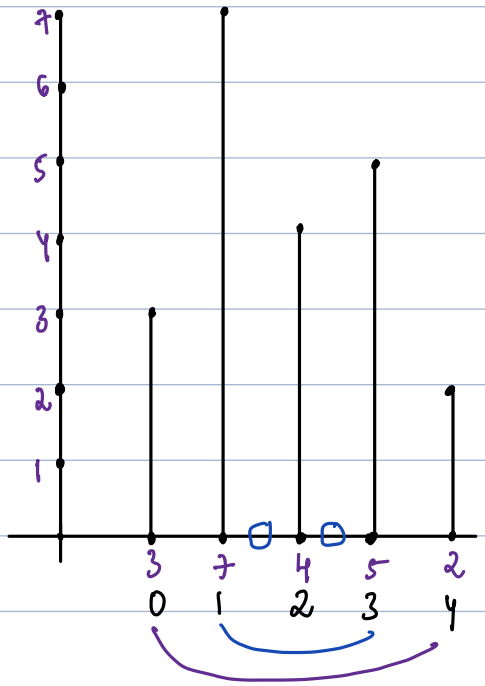
28 Given ar[N] ele, ar[i] represents height of each wall.
Find Max water accumulated between any 2 walls?

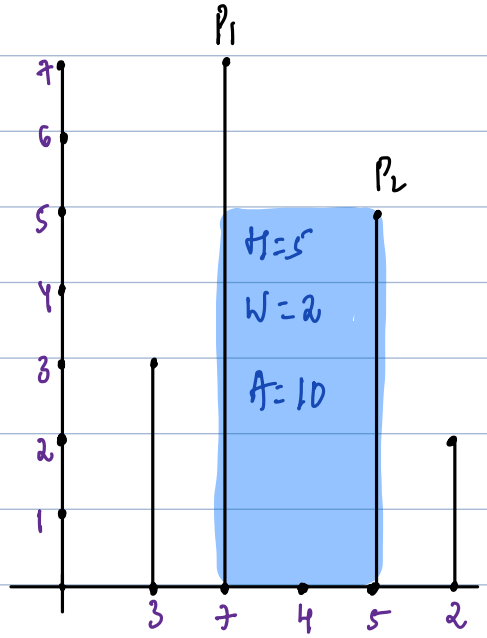
Note1: # Between 2 consecutive walls width = 1

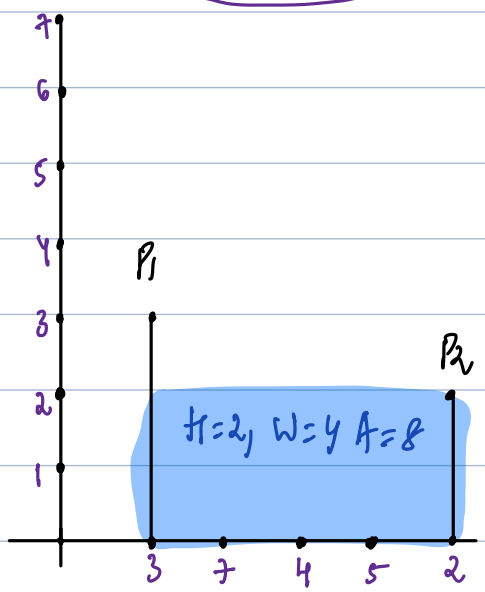
Note2: # When we calculate water between any 2 walls, neglect all other walls between them.

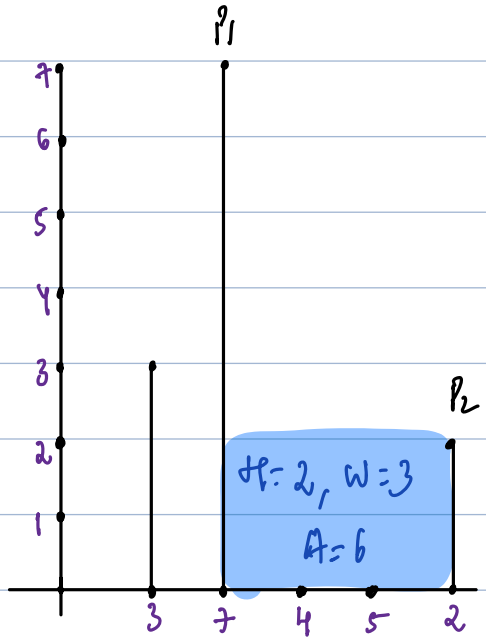
Eg: ar[5] = { 3 7 4 5 2 } ans =

0	1	2	3	4
3	7	4	5	2









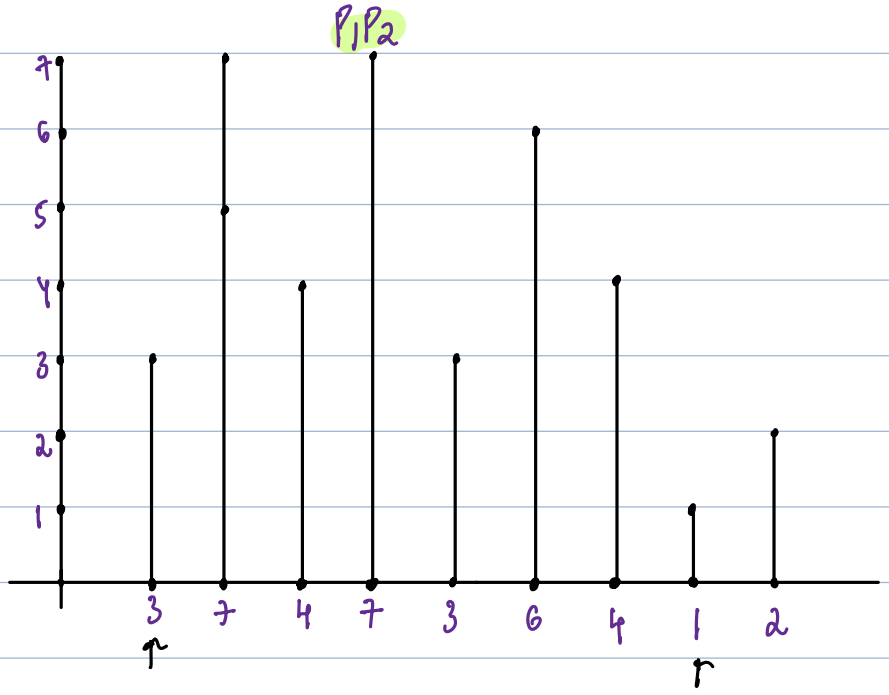
Idea1: Consider every pair of Buildings

```
for(i=0; i<N; i++){        TC: O(N^2)
    for(j=i+1; j<N; j++){
        # (i .. j)
        w = (j-i);   h = min(A[i], A[j])
        Area = w*h
    }
    ans = max(ans, Area);
}
```

Idea2; TC: O(NlogN) TODO

Idea2:

	0	1	2	3	4	5	6	7	8
ar[10] = {	3	7	4	7	3	6	4	1	2 }

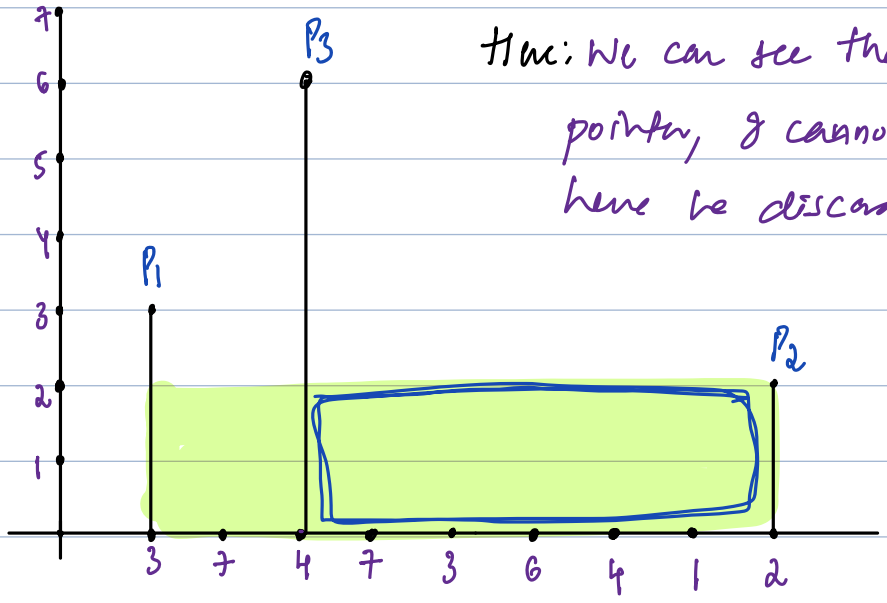


A = 6, 7, 18, 20, 24, 9, 14

P_1	P_2	$H = min(ar[P_1], ar[P_2])$	$W = P_2 - P_1$	Area Update
0	8	H = min(3, 2) = 2	W = 8	16
		H = min( ) =		
		H = min( ) =		
		H = min( ) =		
		H = min( ) =		
		H = min( ) =		
		H = min( ) =		
		H = min( ) =		

while ($P_1 < P_2$)

Q? Which pointer can we update



Hint: We can see that, with P_2 as smaller pointer, I cannot get better area. Hence we discard.

long water(vector<int> &H){

3

Note: In 2 pointer we discard a pointer if with element we cannot get ans a If we cannot get better ans.