17 December 2022 09:21

- → Buy & Sell Stock
- → Target Sum Pair / Triplets / Pair of Roses
- → Product Except Self
- → Book Allocation / Murthal Parantha / Aggressive Cows / Painter's Partition
- → Rain Water Trapping
- → Sorting in linear time
- → Next Permutation
- → 1D, 2D11

A

500

100 ✓

200 X

250 ✓

500 ✓

150 X

$\boxed{3}$

$\underline{\underline{500}}$ 100

200

400 ✓

$\underline{2}$? → True

5 → False

Buy & Sell Stock

0th	1st	2nd	3rd	4th	~~5th~~ ✓	6th	7th	8th	9th	10th
7	$\underline{10}$	5	3 ✓	6	4	2	$\underline{5}$	$\underline{8}$	3	9
x		x	~~x~~	x						

Profit: ~~1~~ ~~1~~ ~~2~~ ~~4~~ ~~5~~ ~~7~~ 8

1) $O(N^2)$ ✓

```
for( int b=0; b<n; b++) {
    for(int s = b+1; s<n; s++) {
        // Comparison Logic
    }
}
```

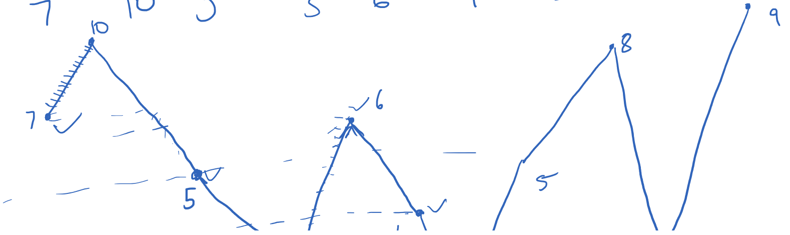
n

n-1, n-2, n-3, ..., 0

$$\frac{(n-1)(n)}{2}$$

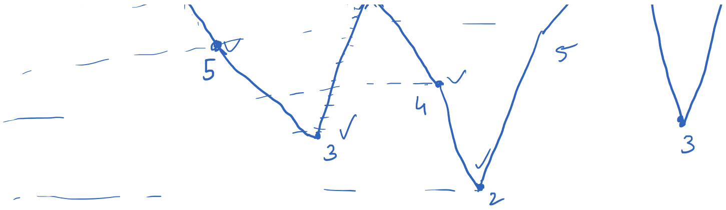
$O(n^2)$

(0) 7 (3) 10 (0) 5 (0) 3 (3) 6 (1) 4 (0) 2 (3) 5 (6) 8 (1) 3 (7) 9



→ Past mein sabse sasta kab tha?

→ Past ki min



Har Bande se pucho
- → Past ki min Value ✓
- → Aaj Bechunga
- → Profit Dekhlunga

```
int ans = 0;
int min = +∞    → Past ki minvalue
for( int s=0; s<n; s++) {
if(min > arr[s]) {
      min = arr[s];
   }
   curr-profit = arr[s] - min;
    if (curr-profit > ans) {
         ans = curr-profit
      }

   }
```

17 December 2022 10:17

Target Sum Pairs

target = 5

| 1 | 3 | 4 | 2 | 5 |

① Generate All Pairs & calculate sum ✓

```
for( f=0; f<n; f++)
   for (s=f+1; s<n; s++){
      curr= arr[f] + arr[s];
      if(curr== targent)
         print pair
}
```

complexity = $\frac{(n-1)(n)}{2}$

$\Rightarrow O(n^2)$ ✓

1 , 4

3, 2 ✓

↳ $f^{(?)} + s^{(?)} = \text{target}$

smaller, larger

② Sorting + Binary Search

| 1 | 2 | 3 | 4 | 5 |

target → 5

✓(?) ✓✓
f + s = target

s = target - (f)
(?)

$O(N^2)$ ← 2 loop

$O(N \log N)$
↳ Merge sort
↳ QuickSort

```
sort (arr);
for( int f=0; f< n-1 ; f++){
      s = target - arr[f];
      if( binary Search (arr, f+1, n-1)) {      (st)  (end)
     ✓   print
      }
}
```

1 → 5-1 = 4

(n-1) times × log n

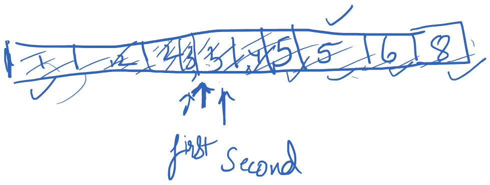
$O(N \log N)$

Time complexity = $O(N \log N) + O(N \log N)$

$= O(N \log N)$

③ Sorting + 2 Ptr

sort (arr);



target = 6

```
while( f < s){

  if (sum = = target){
     print ✓                O(1)
     f++;
     s--;        6 → 5
```

1,5 ✓

2,4 ✓

first second → n elements

2,4 ✓

first + second → n elements visit

O(N)

```
print...
f++;
s--;
}else if (sum > target){
    s--;
} else { // target > sum
    f++
}
}
```

Target Sum Triplets → arr → target

→ i, j, k

① →
```
for(i){
  for(j){
    for(k){
    }
} }
```

$O(N^3)$

② Sort + 2 ptr

$\overset{✓}{f} + \overset{(?)}{s} = \overset{✓}{target}$

$\overset{✓}{f} + \underbrace{\overset{(?)}{s} + \overset{(?)}{t}} = \overset{✓}{target}$

$\underbrace{s + t = target - f}$

$\overset{(?)}{s} + \overset{(?)}{t} = \overset{✓}{target'}$

~~5 7 9 1 2 4 6 8 3~~
10

1 2 3 4 5 6 . 7 8 9

target = 10

target' = 10 − 5 = 5

1,2,7 1,3,6 1,4,5 2,3,5

1) Product Except Self without using division /

n = 4

arr = 1_0 2_1 3_2 4_3 1_4

24 12 8 6 24

O(1)

prefix Sum =

0	1	3	6	10	11
0	1	2	3	4	5

```
for(int i=1; i ≤ n; i++){
    prefix[i] = prefix[i-1] + arr[i-1];
}
```

1_0 2_1 3_2 4_3 1_4 - -

Suffix Sum =

10	8	5	1	0
0	1	2	3	4

5

suffix[i] = suffix[i+1] + arr[i+1];

① Using 2 loops

i)

```
for(int i=0; i<n; i++){
    product = 1
    for(int j = 0; j<n; j++){
        if(i! = j){
            product = product * arr[j];
        }
    }
    syso( product);

}
```

$O(N^2)$

②

```
for(i=0; i<n; i++){

    for( l=0; l<i; l++){
        leftProduct *= arr[l]
    }

    for(r=i+1; r<=n; r++){
        rightProduct *= arr[r];
}   }
```

(i-1)

(n-(i-1))

n x (n-1)

N^2 x

Space Time

Trade Off

Time	Space
O(N)	O(N) ✓
O(N²)	O(1)

1, 2, 3, 4, 1 (indices 0 1 2 3 4)

prefix Product =

1'	1	2	6	24
0	1	2	3	4

suffix Product =

24	12	4	1	1'
0	1	2	3	4

ans = 24, 12, 8, 6, 24

```
for(i=0; i<n; i++){
ans[i] = prePro[i] x sufPro[i];
}
```

O(N)

N

2_0 3 6_2

1'	2	6

18	6	1'
0	1	2

x

O(N)

18, 12, 6

Prefix Product:

```
pre[0] = 1' ;
for(i=1; i<n; i++){
    pre[i] = pre[i-1] * arr[i-1];
}
```

O(N)

Suffix Product:

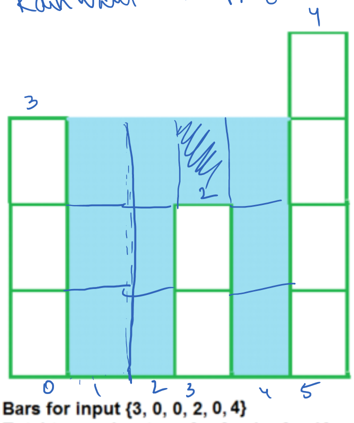
O(N)

```
suf[n-1] = 1'

for(i=n-2; i≥0; i--){
    suf[i] = suf[i+1] * arr[i+1];
}
```

18 December 2022 10:16

Rain Water Trapping



xample : **Bars for input {3, 0, 0, 2, 0, 4}**
Total trapped water = 3 + 3 + 1 + 3 = 10

1st : min (3,4) − 0 = 3

2nd : min (3,4) − 0 = 3

3rd : min(3,4) − 2 = 1

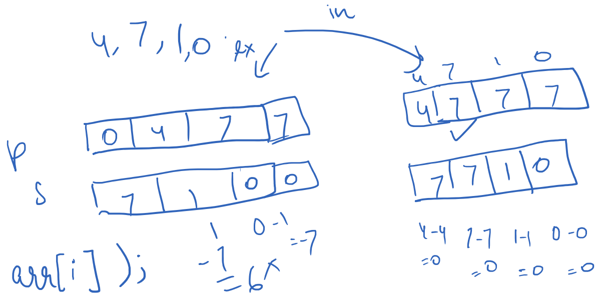
4th : min(3,4) − 0 = 3

?? [0th : min(3,4) − 3 = 0

?? [5th : min(3,4) − 4 = −1?

→ Boundary par nahi hoga pani ;:

Requirements for each building
1) Prefix Max, Suffix Max → calculate minimum
2) arr[i] → Subtract this value



1) Prefix Max → Including self

```
pre[0] = arr[0];
for (i=1; i<n; i++) {
    pre[i] = Math.max (prefix [i-1], arr[i] );
}
```

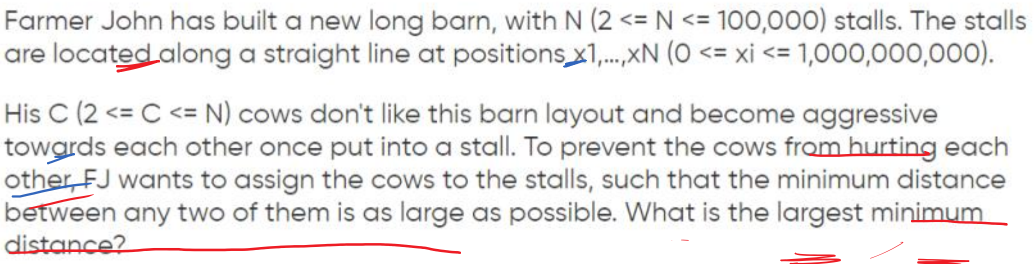
2) Suffix Max → Including self

```
suf [n-1] = arr [n-1];
for (int i=n-2 ; i≥0; i--) {
    suf[i] = Math.max ( suf[i+1], arr[i]);
}
```

3)

```
for (i=0 ; i<n ; i++) {                //prefix
    cw = Math.min ( pre[i] , suf[i]) − arr[i];
    total += cw;
}
print (total);:)
```

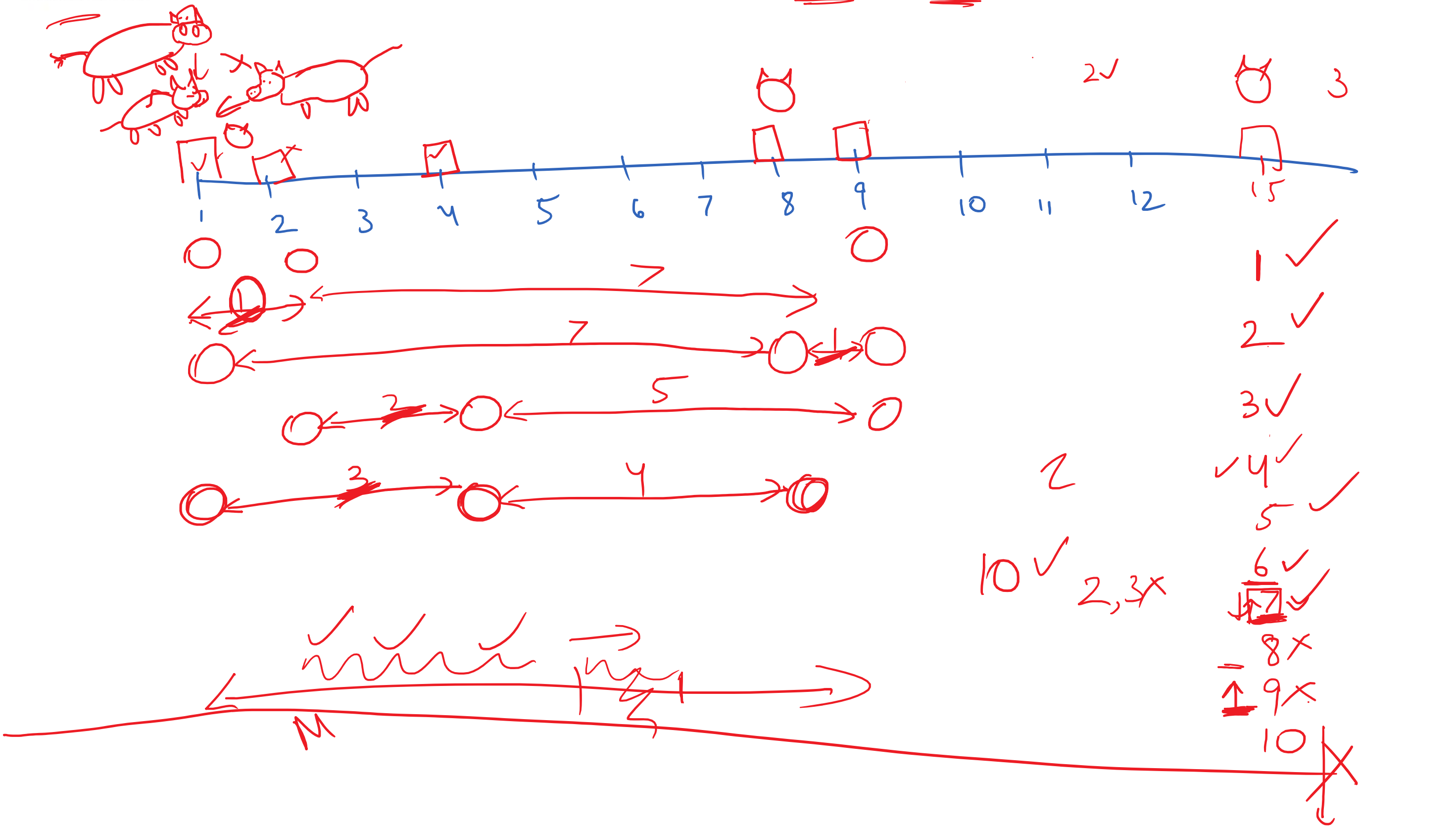
18 December 2022 10:46



Farmer John has built a new long barn, with N (2 <= N <= 100,000) stalls. The stalls are located along a straight line at positions x1,...,xN (0 <= xi <= 1,000,000,000).

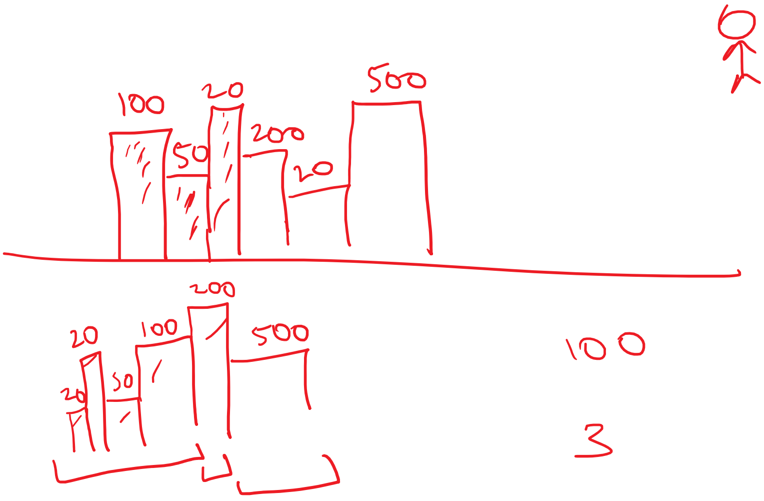
His C (2 <= C <= N) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ wants to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

1 , 2, 4 ,8 9, 15



Maximize the minimum B.S

Minimize the maximum



150

160

100

3