

- 500 100  
200  
400 ✓

500

200 x

Sub ✓

150x

③

2?  $\rightarrow$  Time

$S \rightarrow \text{False}$

0th	1st	2nd	3rd	4th	<del>5th</del>	6th	<u>7th</u>	8th	9th	10th
7	10	5	3	6	4	2	<u>5</u>	<u>8</u>	3	9
+		x	<u>x</u>	x						

1)  $O(N^2)$  ✓

```

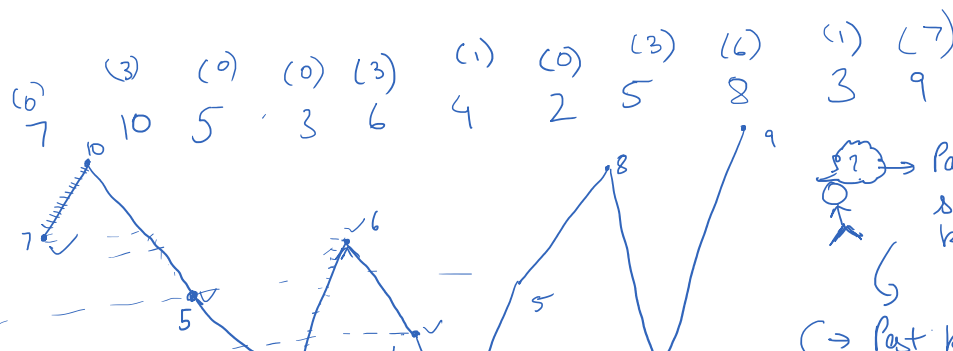
for (int b=0; b<n; b++) {
    for (int s=b+1; s<n; s++) {
        // logic
    }
}


```

## // Comparison Logic

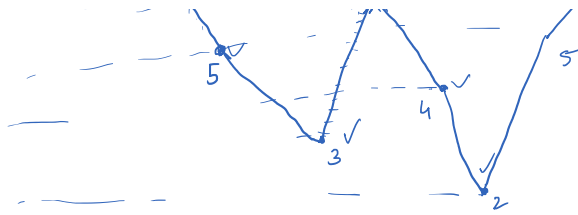
$$n-1, n-2, n-3, \dots, 0$$

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

 $O(n^2)$ 


 Past mein  
 sabse sabta  
 kab tha?

$\Rightarrow$  Post ki min



Har  
 bande  
 se kude

→ Past ki min  
 value ✓  
 → Aaj Bechunga  
 → Profit Dekhunga

```

int ans = 0;
int min = +∞ → Past ki min value
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;
  }
}
  
```

# 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 == target)
            print pair
```

Complexity =  $\frac{(n-1)(n)}{2} \Rightarrow O(n^2)$  ✓

1, 4  
3, 2 ✓  
f(?) + s(?) = 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  
↳ Quick sort

```
sort(arr);
for (int f = 0; f < n - 1; f++) {
    s = target - arr[f];
    if (binarySearch(arr, f + 1, n - 1, s))
        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 Pter

sort(arr);

1 | 2 | 3 | 4 | 5 | 5 | 6 | 8  
↑ ↑  
first second

1, 5 ✓  
2, 4 ✓

but we need → n elements

target = 6  
while (f < s) {  
 if (sum == target) {  
 print ✓  
 f++;  
 s--;  
 }  
}

2,4 ✓

first + second  $\rightarrow n$  elements visit

$O(N)$

```

    while (f < s) {
        if (sum < target) {
            sum += arr[f++];
        } else if (sum > target) {
            sum -= arr[s++];
        } else {
            // target == sum
            return true;
        }
    }
    return false;
}

```

Target Sum Triplets  
 $\rightarrow i, j, k$

$\rightarrow arr$   
 $\rightarrow target$

①  $\rightarrow$  for(i) {  
     for(j) {  
         for(k) {  
             // ...  
         }  
     }  
 }

$O(N^3)$

② Sort + 2 ptr

$f + s = target$   
 $f + s + t = target$   
 $s + t = target - f$   
 $s + t = target$

5 7 9 1 2 4 6 8 3  
 10

1    2    3    4    5    6    7    8    9

target = 10 - 5 = 5

target = 10

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