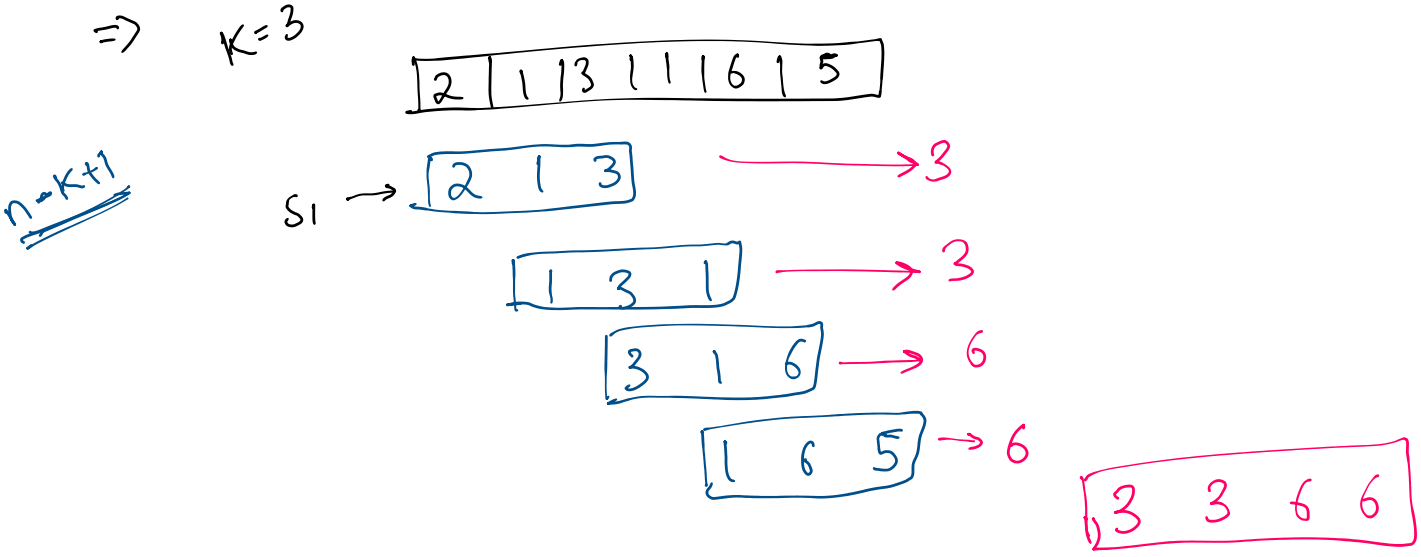
Problem Solving Class
(for sliding window)

Que.1 Find the maximum element of all subarrays of size k.

=> k=3

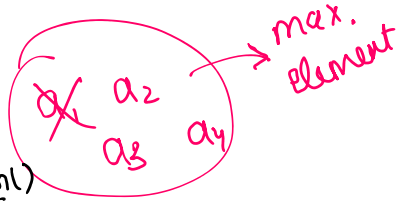


$ans[S_1] = max(a_1, a_2, a_3)$

$ans[S_2] = max(a_2, a_3, a_4)$

$ans[S_3] = max(a_3, a_4, a_5)$

O((n-k+1) * k)



O(n log K)

*st.begin()

sorted { 1, 2, 3 }

multiset

map

{1,2,3}

{{1,1},{2,1},{3,1}}

{{3,1}}

follow up :-

O(N)

(*mp.begin()).first

standard technique

n=7, K=3

5	1	4	3	6	4	7

5 4 → {5,4}

4 3 → {4,3}

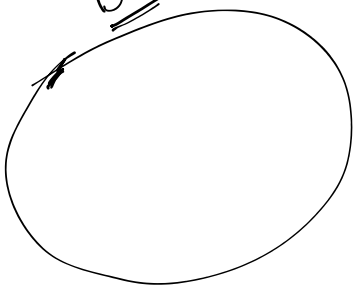
6 → {6}

6 4 → {6,4}

7 → {7}

FIFO

DS



K=3

5	4	3	4	1

5 4 3 → {5,4,3}

4 4 → {4,4}

data structure will have elements in decreasing order

→ {4, 1}

insert (back) → pop_back () ✓
push_back () ✓

remove (front) → pop_front () ✓

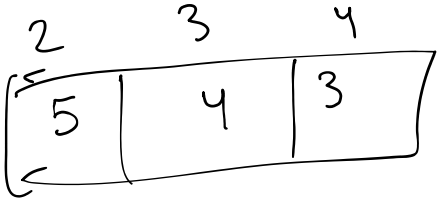
stack ✗
queue ✗
vector ✗
deque ✓

tell me the largest guy.

✓ front() front of the data structure

interviews

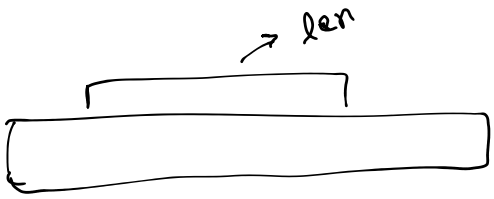
pair < int, int >
↓ value ↓ index



{{5,2}, {4,3}, {3,4}}

Que. maximum sum of any subarray with length i (... a and b.

Que. maximum sum of any subarray with length b/w a and b.



len ≥ a

&& len ≤ b

=>

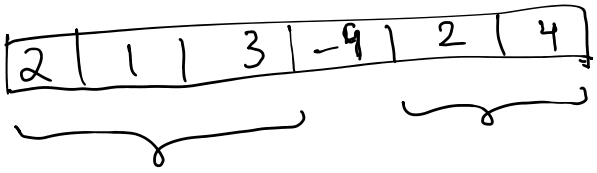
$O(n^2)$

$O(n * k)$ (n, k)

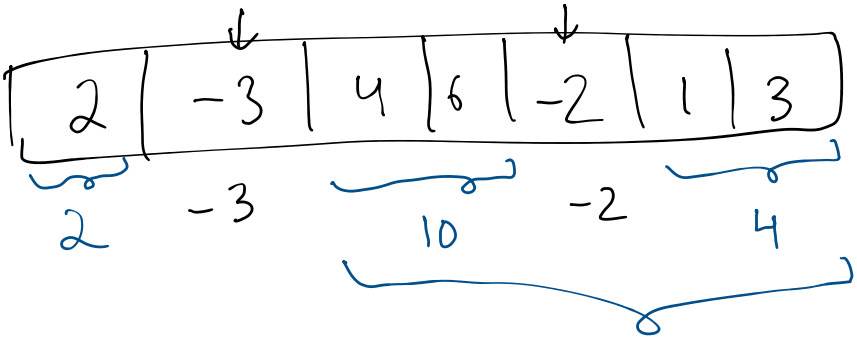
$O(n^2)$

```
int ans = 0
for (K = a; K ≤ b; k++) {
    max_sum = -1e9
    ⋮
    ans = max (ans, max_sum)
}
```

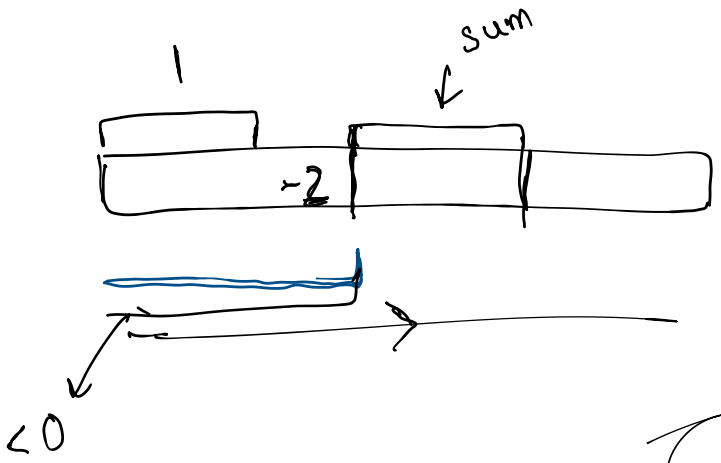
maximum subarray sum



6 6

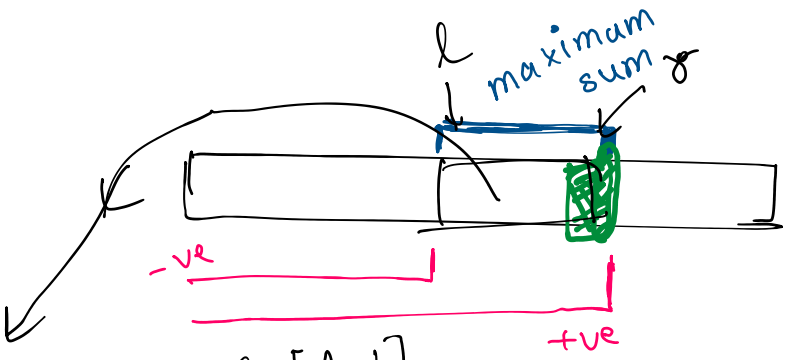


12

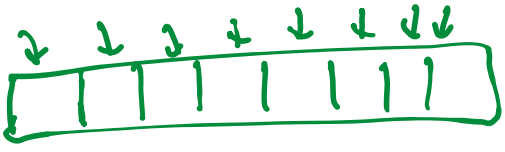


Kadane's Algorithm

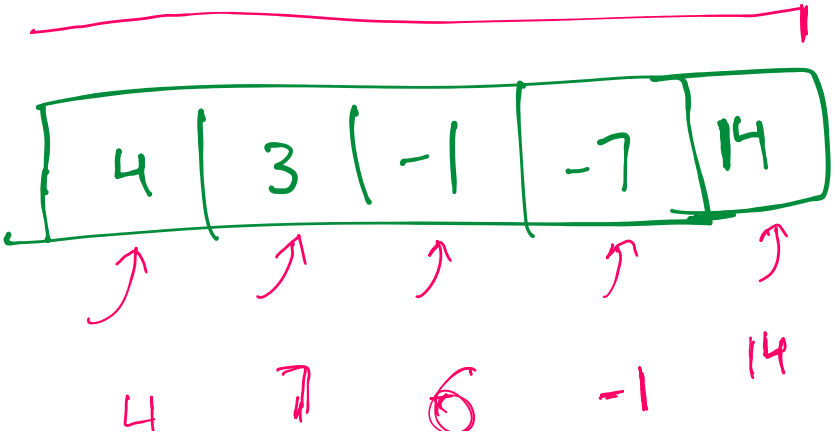
H.W



prefix[r] - prefix[l-1]

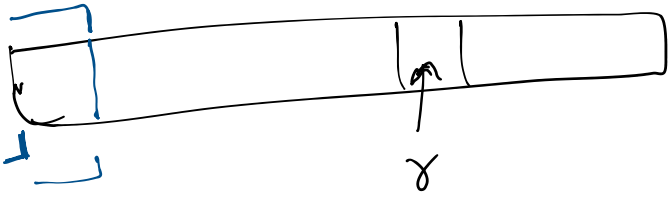


ans



4 7 6 -1 14

pre[i] → sum of first i elements (1-based indexing)



pre[0] = 0

$$\max\left(\max_{1\le l\le r}\left(pre[r] + (-pre[l-1]),\ 0\right)\right)$$

$$\Rightarrow pre[r] - \min_{1\le l\le r}\left(pre[l-1]\right)$$

$$\Rightarrow pre[r] - \underbrace{\min_{1\le l\le r}\left(pre[l-1]\right)}_{\text{min\_prefix}}$$

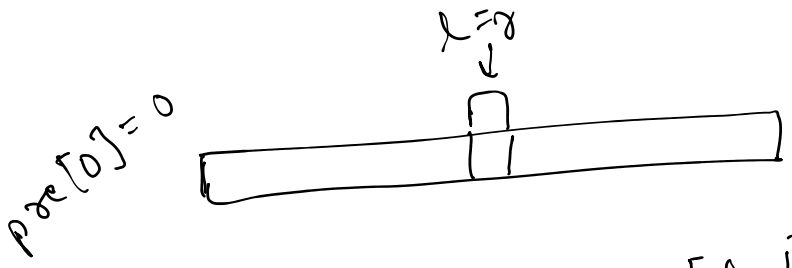
pre[r] → current_prefix

1	2	-6	3	4	-5	6
0	1	3	-3	0	4	-1	5
	1	3	0	3	7	2	8

sum = 0, min_sum = 0, ans = 0;

```
sum = 0, min_sum = 0, ans = 0;
for (i = 1; i <= n; i++) {
    sum += a[i]
    min_sum = min(min_sum, sum)
    ans = max(ans, sum - min_sum)
}
```

max. subarray sum of any subarray with atleast mn elements.



$r = l \rightarrow 1$

$len = r - l + 1$

$r - l + 1 \geq mn$

$l \leq r - mn + 1$

$\max(pre[r] - pre[l-1])$

$1 \leq l \leq r - mn + 1$

$1 \leq l \leq 1$

$r == mn$

$pre[l-1] = 0$

```
min_prefix = 10^9, sum = 0, ans = -10^9
for (i = 1; i <= mn; i++) sum += a[i]
min_prefix = 0
ans = sum - min_prefix
```

ans = sum − ...

. r++) {

mn = 3
mx = 5

lmn lmn = 3 lmx lmx = 5 r = 6 r = 7

| 1 | 2 | -6 | 3 | 4 | -3 | 5 | 7 |

0 1 3 -3 0 4 1 6 13

mx − mn + 1

lmn − 1 to lmx − 1

pre[r]

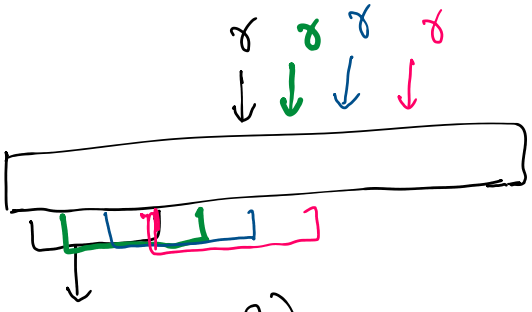
min pre[l − 1]

lmn ≤ l ≤ lmx

len ≥ mn

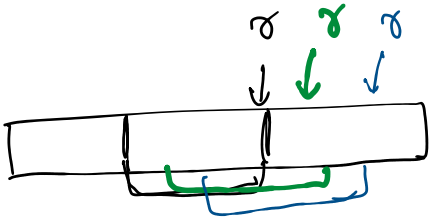
l ≤ r − mn + 1
lmx = r − mn + 1
lmn = r − mx + 1

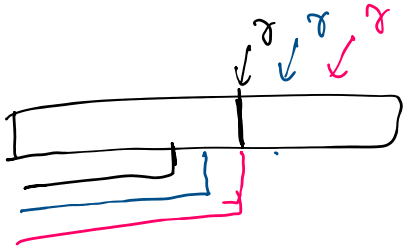
len ≤ mx
r − l + 1 ≤ mx
l ≥ r − mx + 1



min (pre[i])

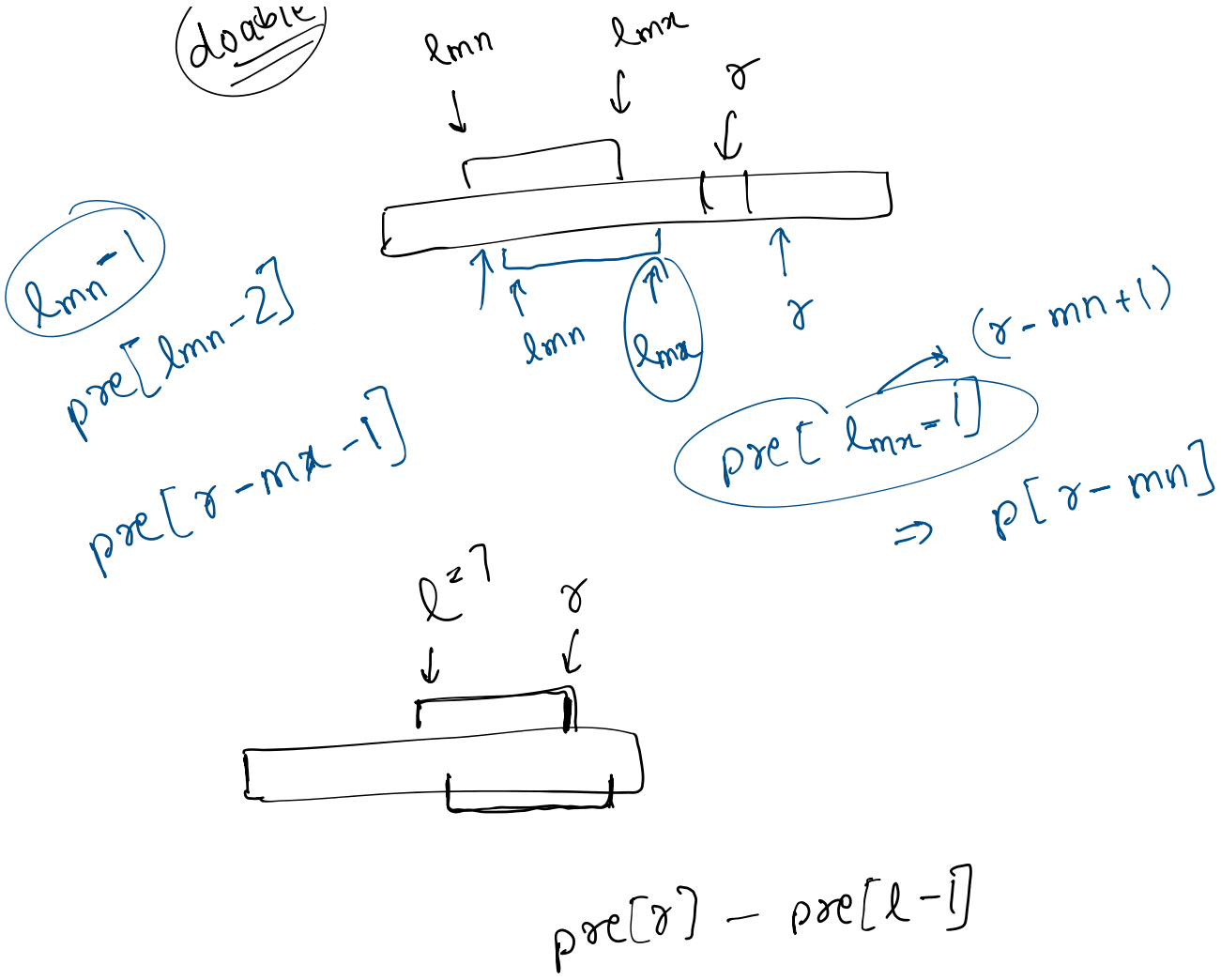
mx − mn + 1





doable

lmn lmx r



1	2	-3	4	-5

H.W :-

1. Do all problems of previous class

2. Try to do them in O(N) also

3. DO today's problem

(6) LJD today's problem

DPPs

Code - https://hastebin.com/share/ajufokunin.cpp