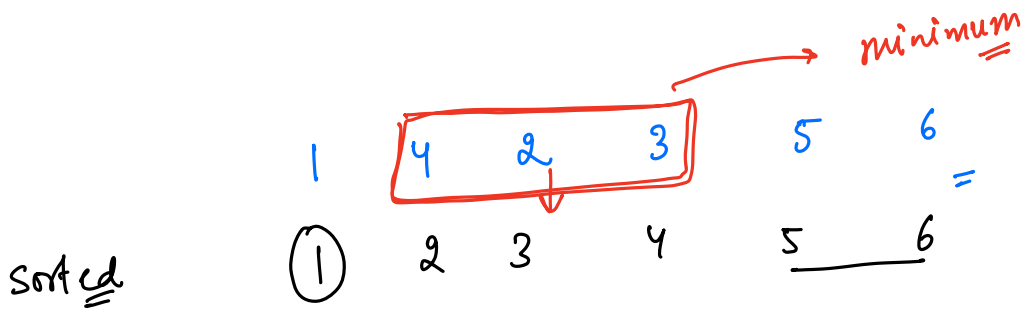
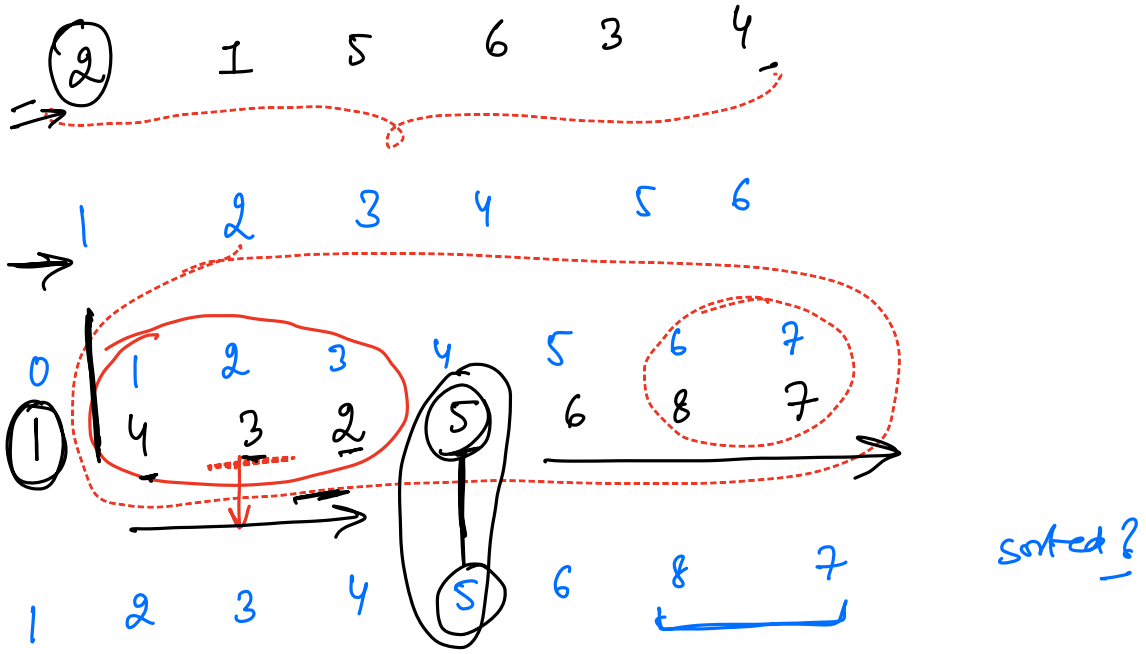
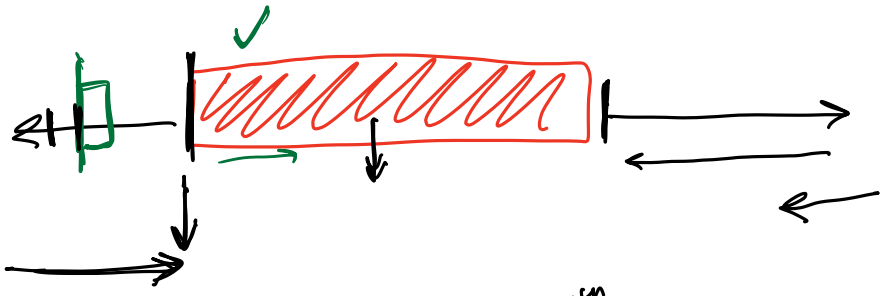
Problem Solving 3

① You are given with an array.
Find minimum length subarray such that if you sort that subarray, complete array would be sorted.





idea:



Dumy → Sort that array

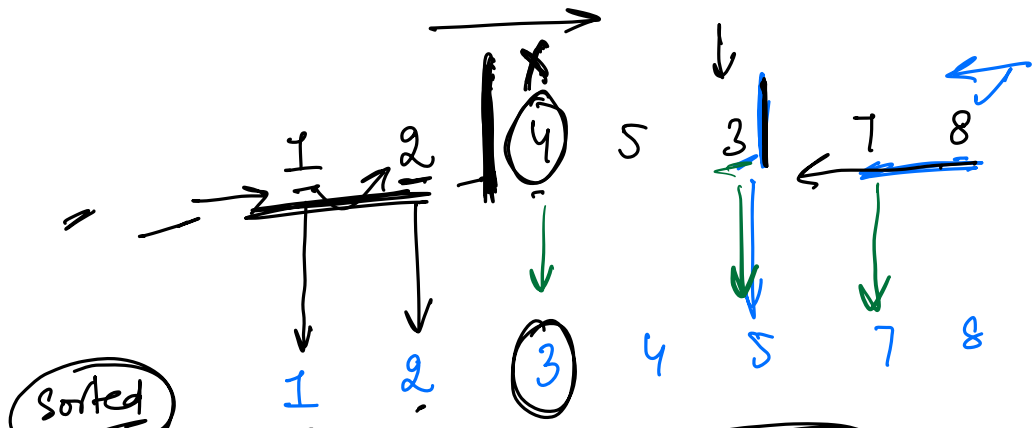
i^{th}

j^{th}

start

subarray

end



arr[i] == sorted[i]

$0 < N < 10^9$

T.C: $O(N)$

T.C: $N \log N$ (comp) S.C: $O(N)$

1) Build another array — sorted
2) start comparing from begin stop when found — i violating condition
3) Back to — j

Q2 M puzzles → N puzzles

M → every puzzles has certain no of pieces

A: max no. of pieces of any puzzle

B: minimum no of any piece

$\boxed{\text{minimize } (A - B)}$

① 10 15 6 9 13 8

M = 6
N = 4

① 15 6 9 13 → A = 15, B = 6 → 9

10 15 6 9 → A = 15, B = 6

10 6 9 8

A = 10

B = 6

4

15 13

10 9 8 — 13 ✗

1 sort & pick first N elements

N = 4

1 2 3 110 111 112 113

109 3

1 2 3 4 110 111 112

sorted

0th 3rd

6 8 9 10 13 15

B B A A

A

4

a th → i + n - 1 th

i B A

1) sorted array

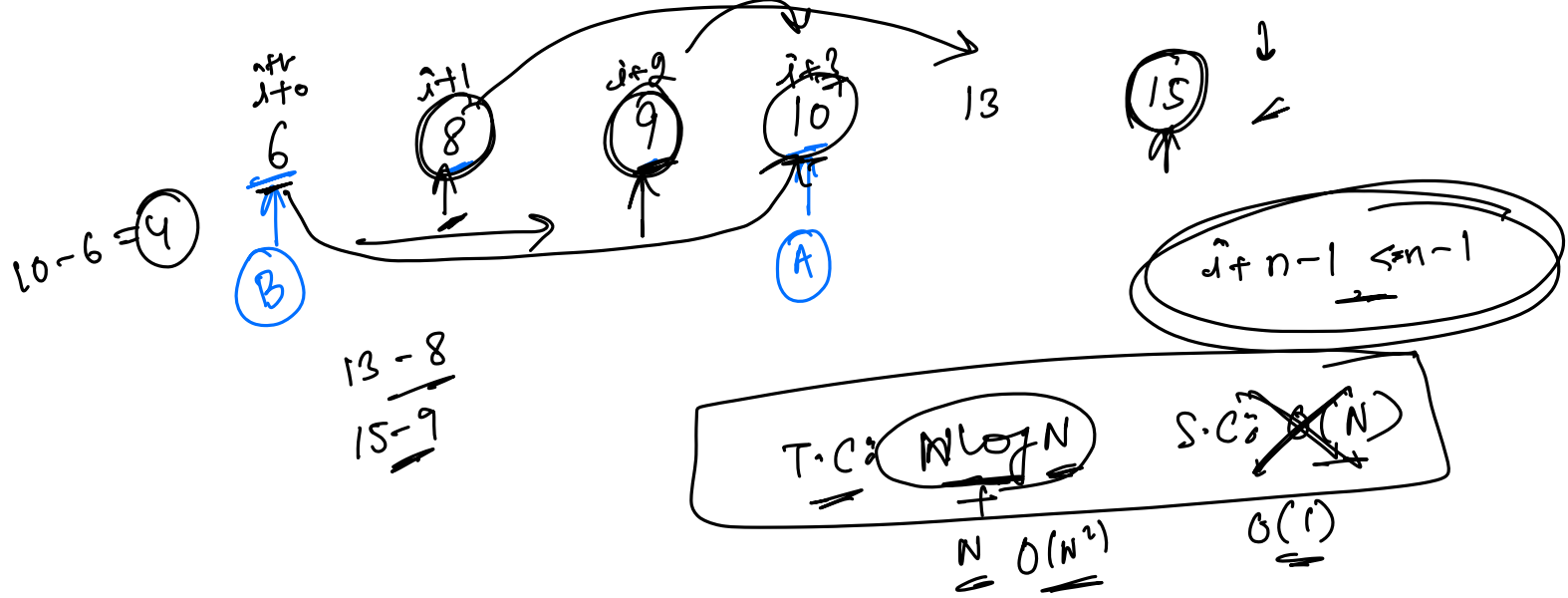
i = 0 i++

diff a[i+n-1] - a[i]

ans — smaller update

i th i+1 i+2 i+3 ... i+n-1

B

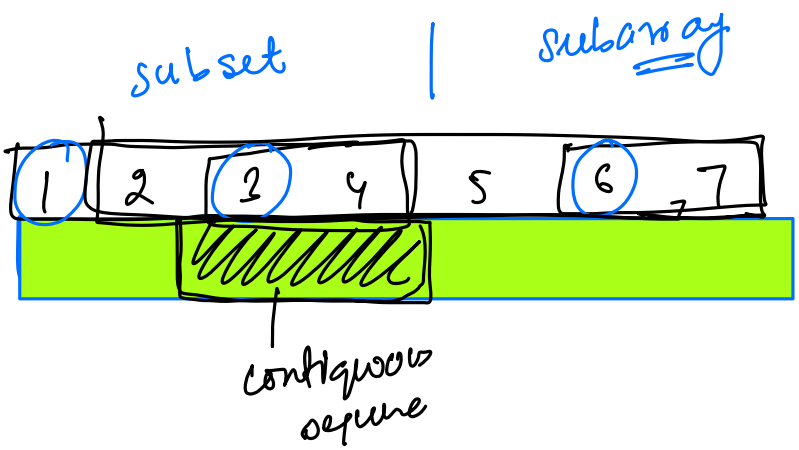


(3) given with array of N elements →
split this array into two non-empty subsets

(A) (B)

subsets / subarray ✗

minimize $|\max(A) - \min(B)|$



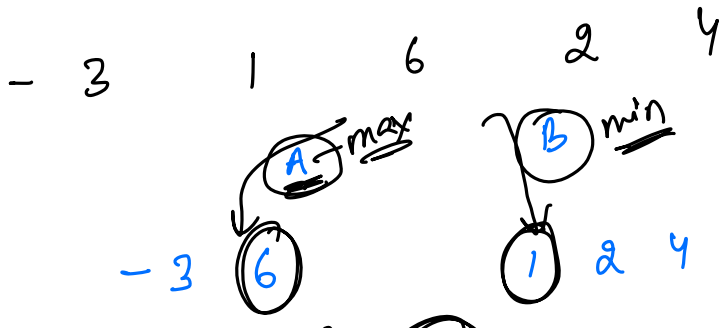
subset

{3 6 1}

subsequence

3 6 1 ✗

1 3 6 ✓



diff 5

2 4 1 -3 6

4 2 7 -3

1 2 4 -3 6

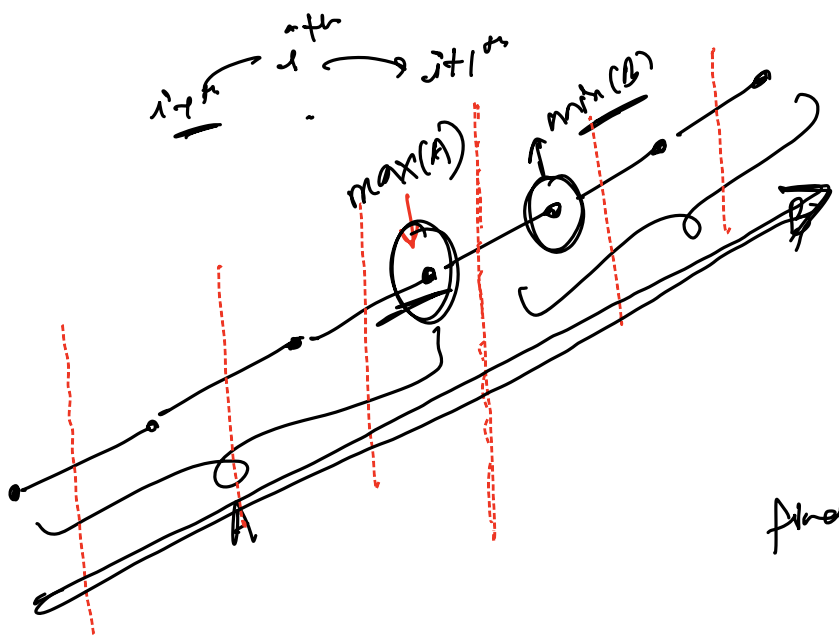
-3 1 2 4 6

|1-2| = 1

sort the array

prev grad 9 → next grad

after sorting → min diff is formed from adjacent elements



We'll be consider difference among all adjacent elements

↓

find ans → min diff

INT_MAX infty

ans = 0

1 4 2 4 6

i=0 i=1 i=2 i=3 i=n

-3 1 2 4 6

1 2 2

$|(-3)-(1)|$
$= 4$

(1) sort the array

$i = 0$

$ans = \infty$

$ans < |arr[i] - arr[i+1]|$

update ans

$i++$

$i == n-1$

break

cout << ans

(4) N elements — 0's or 1's

maximum length subarray → it contain equal no of 1's and equal no of 0's

0	1	2	3	4	5	6
1	0	1	0	1	1	0

(B-F) → consider all possible subarrays

start: $i \to 0 \to n-1$

end: j, $i \to n-1$

3

0 ← (0,0)

```
for( int i=0; i<n; i++)
{
    for( int j = i; j<n; j++)
    {
        [i — j]    c1=0, c2=0,
        for( int k=i; k<=j; k++) }
```

$O(n^3)$

$O(n^2)$

no_of's

Prefix sum

Subarray length $\leftarrow x$

1	1	0

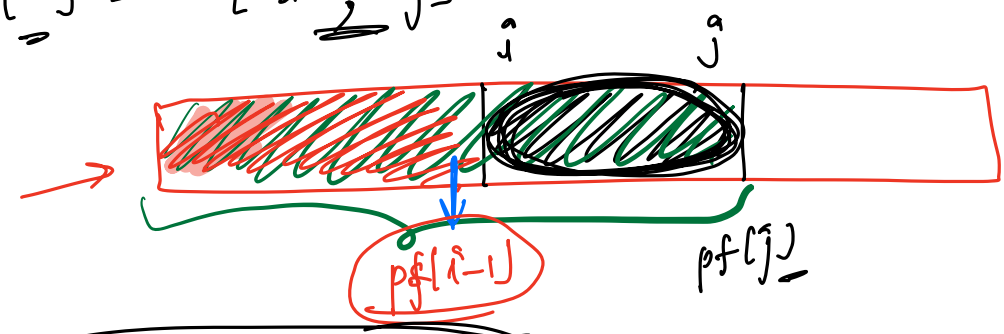
Sum $= x/2$

$a + b = x$

$a + a = x$

$a = x/2$

prefix[ ] = [i , j]



$sm[i-j] = pf[j] - pf[i-1]$

$j - i + 1$

$\rightarrow O(n^2)$

$O(n)$

1 $\longleftarrow$ -1

after conv every 0 to -1

longest subarray with sum=0

1 0 1 0 1 1 0

suba [i,j]

1	-1	1	-1	1	1	-1

$\rightarrow 0$

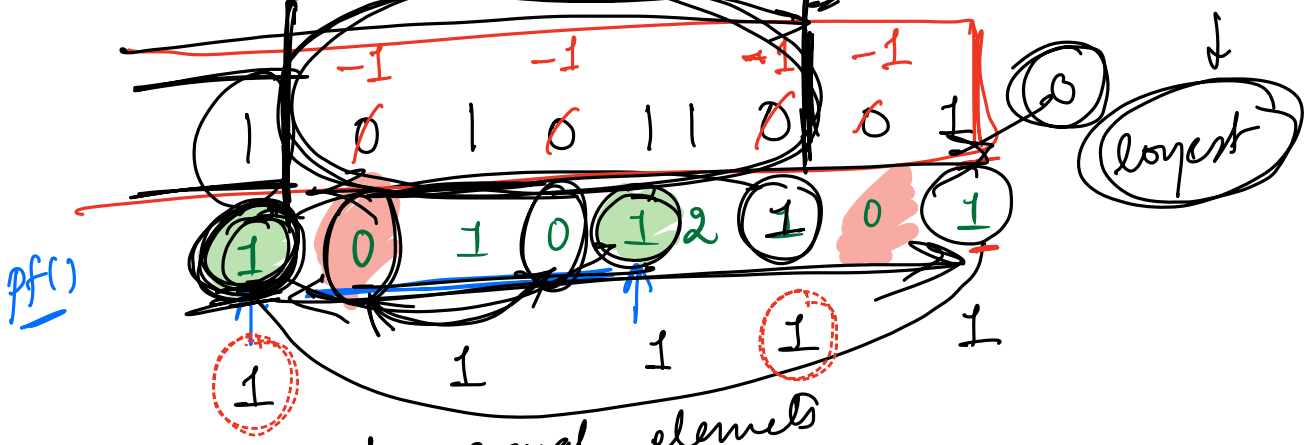
pf[i-1]

pf[j]

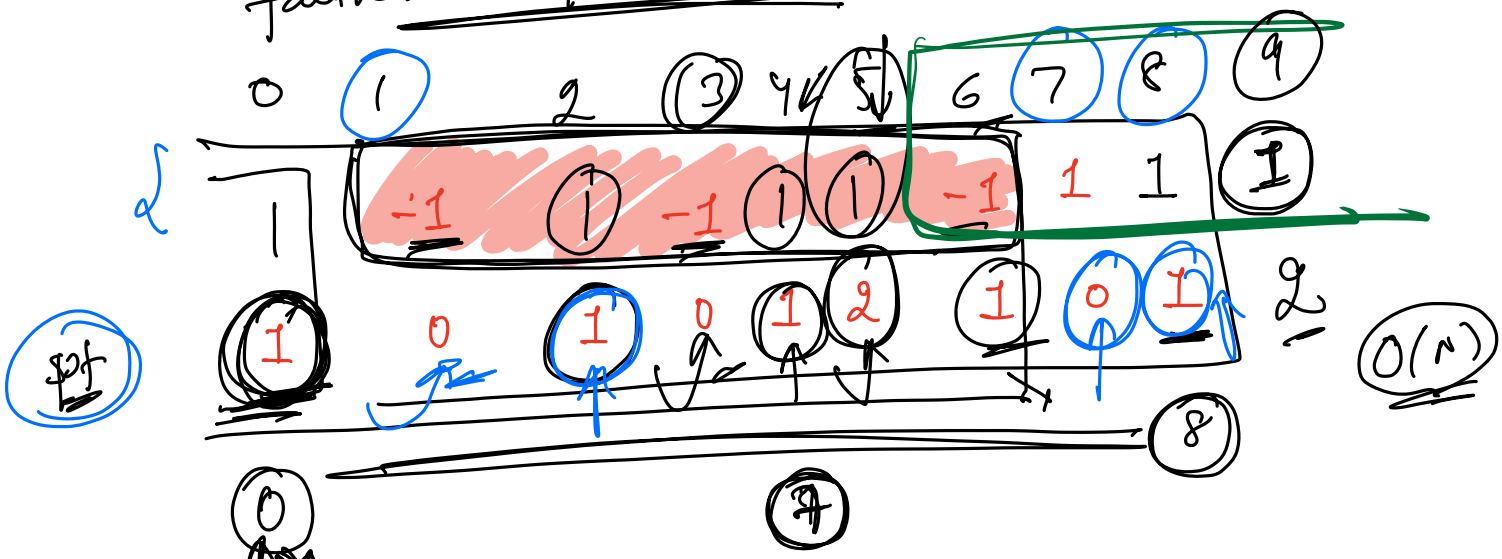
$i - j$

$pf[j] - pf[i-1] = 0$

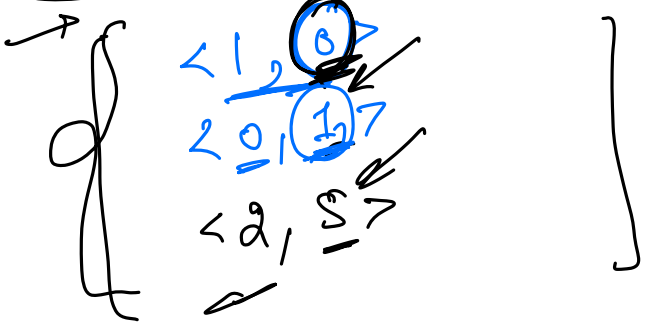
$pf[j] = pf[i-1]$



farthest equal elements



hashmap → value → index

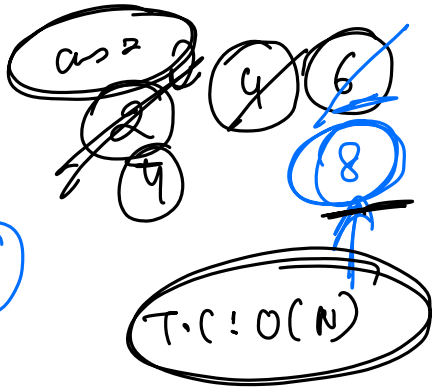


0+1 → 2

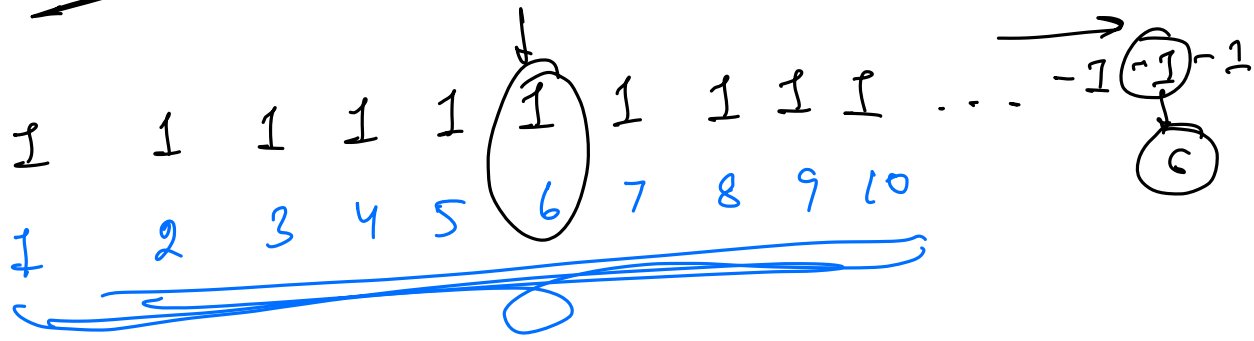
7 − 1 = 6

8 − 0 = 8

9 − 5 = 4



key, first index
value



(5) construct an array with the given info →

- size = k
- elements should be sorted (↑ order)
- positive (>0) and distinct elements
- diff b/w all pair of adjacent elements shoul be equal

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6$

$a_2 - a_1 = a_3 - a_2 = a_4 - a_3 = a_5 - a_4 = a_6 - a_5$

- x, y : $x < y$

largest value of the array is as small as possible

no of eles

$k = 5$ $\quad x = 20 \quad y = 50$

max = ∞

~~max = INT_MAX~~

~~140~~ ~~80~~ 50

1) 20 50 80 110 140 $\quad$ max = 140

2) 20 35 50 65 80 $\quad$ max = 80

3) 10 20 30 40 50

$k = 4 \quad x = 20 \quad y = 40$

10 20 30 40

$k = 4 \quad x = 2 \quad y = 3$

0 1 2 3

1 2 3 y → max

BaP

$a \quad a+d \quad a+2d \quad a+3d \quad \ldots \quad a+(k-1)d$

$a = \{1, x\}$

$d = \{1, y-x\}$

$> x$

4 6 10

$x \ldots y$

$y-x$

x

max = INT_MAX ∞

```
for( int a = 1; a <= x; a++)
{
    for( int d = 1; d <= y-x; d++)
    {
        for( int i = 0; i < k; i++)
        {
            temp[i] = a + i*d;
        }

        if( x is there && y is there &&
            temp[k-1] < max)
            max = temp[k-1];
            ans = temp;
    }
}
}
```

generate AP for a & d

T.C

$O(x*(y-x)*k)$

$a \quad a+d \quad a+2d \quad a+3d \quad \ldots \quad a+(k-1)d$

$x \qquad y$

$x = a + p1 \cdot d$

$y = a + p2 \cdot d$ $\qquad p1 \le p2$

$y - x = (p2 - p1) * d$ → the diff will be factor $(y-x)$

$(p2-p1)$ is

$a+p1d \quad d \quad d \quad d \quad d \quad d \quad d \quad a+p2d$

$x \qquad (p2-p1) \qquad y$

$a + p1d + (p2-p1) * d$

$(p2-p1)+1$ elem

$$\boxed{\text{Total elem} = \left(\frac{y-x}{d}\right) + 1}$$

$k=6 \qquad x=3 \qquad y=9$

$y - x = 6$

1 2 3 6 } d

$d=1 \qquad \frac{y-x}{1} + 1 = 7$

$d=2 \qquad \frac{6}{2} + 1 = 4$

um = 2

1 3 5 7 9 11

$d = 3$ $\qquad \frac{6}{3} + 1 = 3$

How to calculate factors?

32

57

1	*	32	= 32
2	*	16	= 32
4	*	8	= 32
8	*	4	= 32
16	*	2	= 32
32	*	1	= 32

factors come in pair + after a certain point factors start repeating

$x * y = N$

$x <= y$

$x > y$

$x = y$

$x * x = N$

$x <= \sqrt{N}$

factor

```
for( int i=1; i<=√N; i++)
{
    if( N % i == 0)
    {
        factor.push_back(i);
        // i * N/i = N
        if( i != N/i)
        factor.push_back(N/i);
    }
}
```

$O(\sqrt{N})$

Sort the factors

25

$1 \times 25 = 25$

$5 \times 5 = 25$