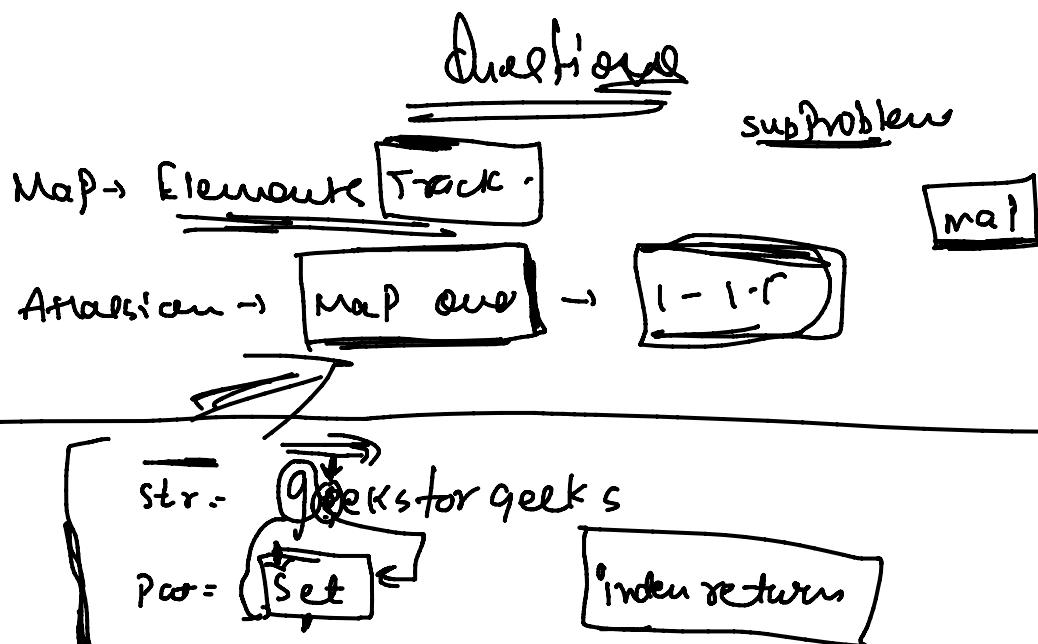


CLASS - 20

- HashTable
- Hash Functions
- Collision Handling
 - Separate chaining
 - Double Hashing
 - closed "
- Relinking
- ordered-map vs map* vs multimap vs ordered_multimap
- ordered-set vs set vs multiset vs ordered_multiset
- customized comparators
 - functors
 - function pointers
 - lambda Expressions
- " Hash functions

Brute force

```
For (i=0; i<str.size(); i++)
```

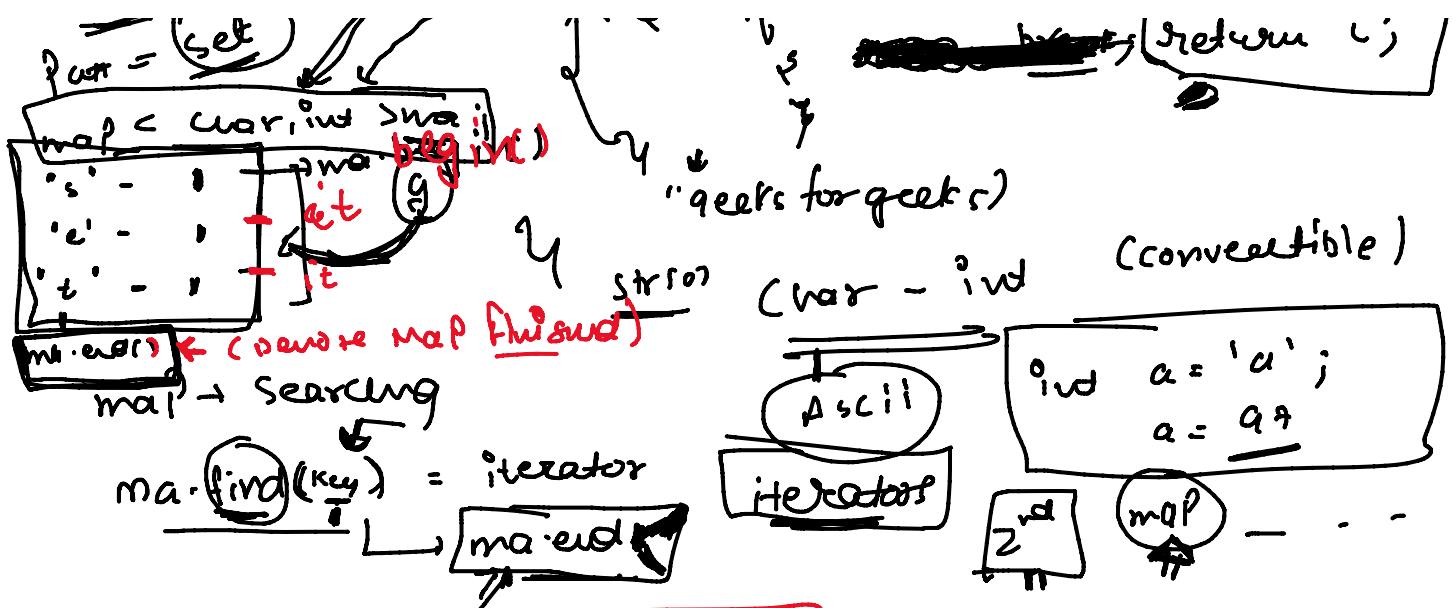
set str[i]

```
for (j=0; j<Parr.size(); j++)
```

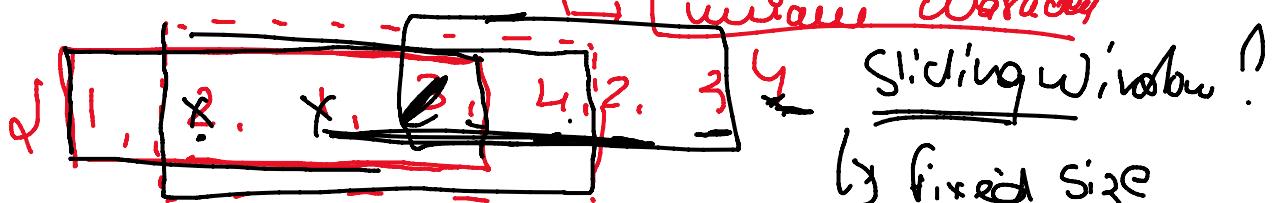
if (str[i] == Parr[j])

return i;

map
Parr = set



Map Sliding window into of window store = ?
 → No of characters unless window



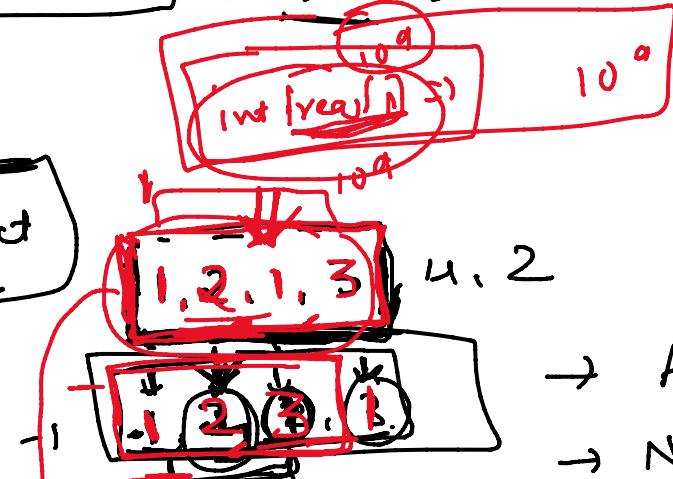
$K=4$
 $1 \rightarrow 3 \quad 3 \rightarrow 1$ $5^{\text{th}} \rightarrow 3$ Window
 $2 \rightarrow 4 \quad 4 \rightarrow 3$
 $\Rightarrow 3, 4, 4, 3$

Normal Approach

Count! Distinct

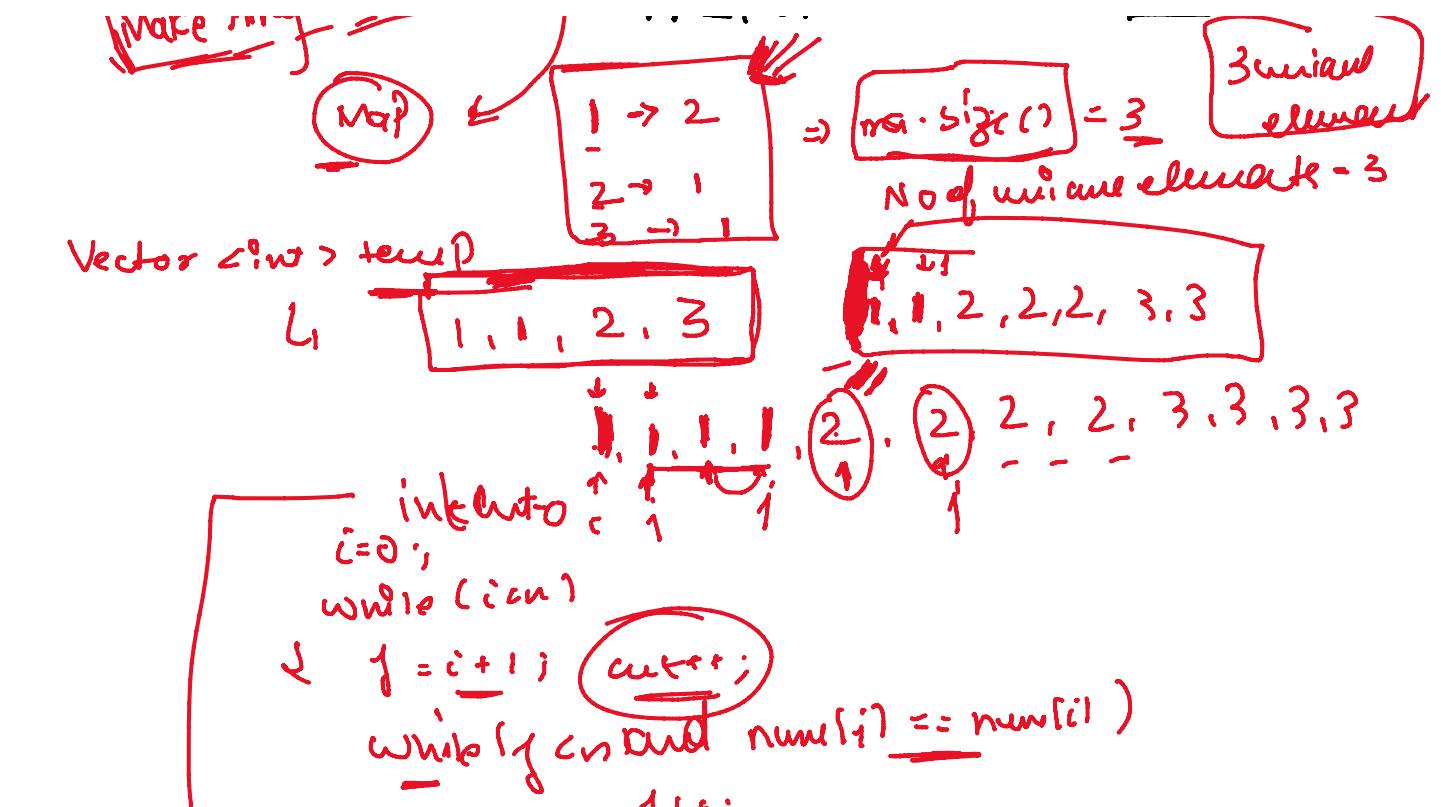
1, 2, 1, 3

Make Array Unique



1, 2, 3, 1

Binary



~~main() +~~ main() = {main();} + -
unordered_map<int,int> mai;

```

for (i := 0; i < k; i++)
    ma[new[i])++;

```

Vector<int> ans;

ans.push_back(ma.size());

$F_{02} \quad (i := k; \ i < n; \ i++)$

ma

3

4

3

$ma \cdot \sigma(1) = 3$

ma · Siglo

$\text{For } (i = k; i < n; i++)$
 ↳ $\underline{\text{ma}[\text{num}[i]]}++;$
 $\underline{\text{ma}[\text{num}[i-k]]}--;$
 {
 $\underline{\text{if } (\text{ma}[\text{num}[i-k]] == 0)}$
 $\text{ma}.\text{erase}(\underline{\text{num}[i-k]})$
 }
 $\text{ave.push_back}(\underline{\text{ma}[\text{size}]})$

5 - 0
2 0 4

$$A = \{ \underline{1}, \underline{3}, 2, 1, 1, 2 \} = \{ 3 \}] \quad 1$$

$$B = \{ \underline{1}, 6, 3, 3 \} = \{ 15 \}$$

Brute force

Goal → 2 element sum

$\text{For } (i = 0; i < n; i++) \rightarrow A$

a b

$\text{for } (j = 0; j < m; j++) \rightarrow B$

$a = \underline{\text{num}[i]}$;

$b = \underline{\text{num}[j]}$;

if ($s_1 - a + b = s_2 - b + a$)

a, b

Brute force
we will make
all sum

Pair

$$\frac{A}{B} \rightarrow \frac{s_1 - a + b}{s_2 - b + a}$$

$$\begin{aligned}
 & \text{Given } B \rightarrow S_2 - b + a \\
 & S_1 - a + b = S_2 - b + a \\
 & S_1 + 2b = S_2 + 2a \\
 & S_1 - S_2 = 2(a - b) \\
 & a - b = \frac{S_1 - S_2}{2} \\
 & \text{array } \rightarrow \text{integer}
 \end{aligned}$$

Pair

A \rightarrow B

ind \downarrow 11 decimal Int

= decm

Map

- if $((S_1 - S_2) \mod 2 == 0)$

return -9; "fair won't exist"

$\frac{S_1 - S_2}{2}$

\downarrow decm

~~Q: $S_1 - S_2 + b$~~

for ($i = 0; i < n; i++$) $\rightarrow B$ array

int dif = $(S_1 - S_2)/2;$

for ($j = 0; j < m; j++$) $\rightarrow A$ array

~~if (arr[i] == dif + arr[j])~~

return $\langle a, b \rangle$;

~~if (arr[i] == dif + arr[j])~~

+ array contains

dif + b

array will element

map::find(dif + b)

unordered_map<int, int> ma;

for (auto i : A)
ma[i] ++;

$S_1 \rightarrow A$
 $S_2 \rightarrow B$

$\text{int diff} = (s_1 - s_2) / 2$
 if ($(s_1 - s_2) \% 2 \neq 0$) return -1;
 \downarrow

for (auto b : B)
 \downarrow if ~~(ma.find (diff + b))~~
 return ~~{diff + b, b}~~;
 \downarrow

a, b

Two Pointer $\downarrow^i \downarrow \downarrow$
 $A \rightarrow [0, 1, 1, 2, 4] = 11$
 $B \rightarrow [3, 3, 3, 6] = 15$



1 - 6

= -5

\downarrow^3

$\textcircled{1} - \textcircled{2} = -4$

$= -2$

$1 - 3 = -2$

-4

if $(a[i] + b[i]) < \text{target}$

$i++$

$a[i] + b[j]$

else if $(a[i] + b[j]) > \text{target}$

$$a[i] - b[j] = \frac{s_1 - s_2}{2}$$

$i++$

target

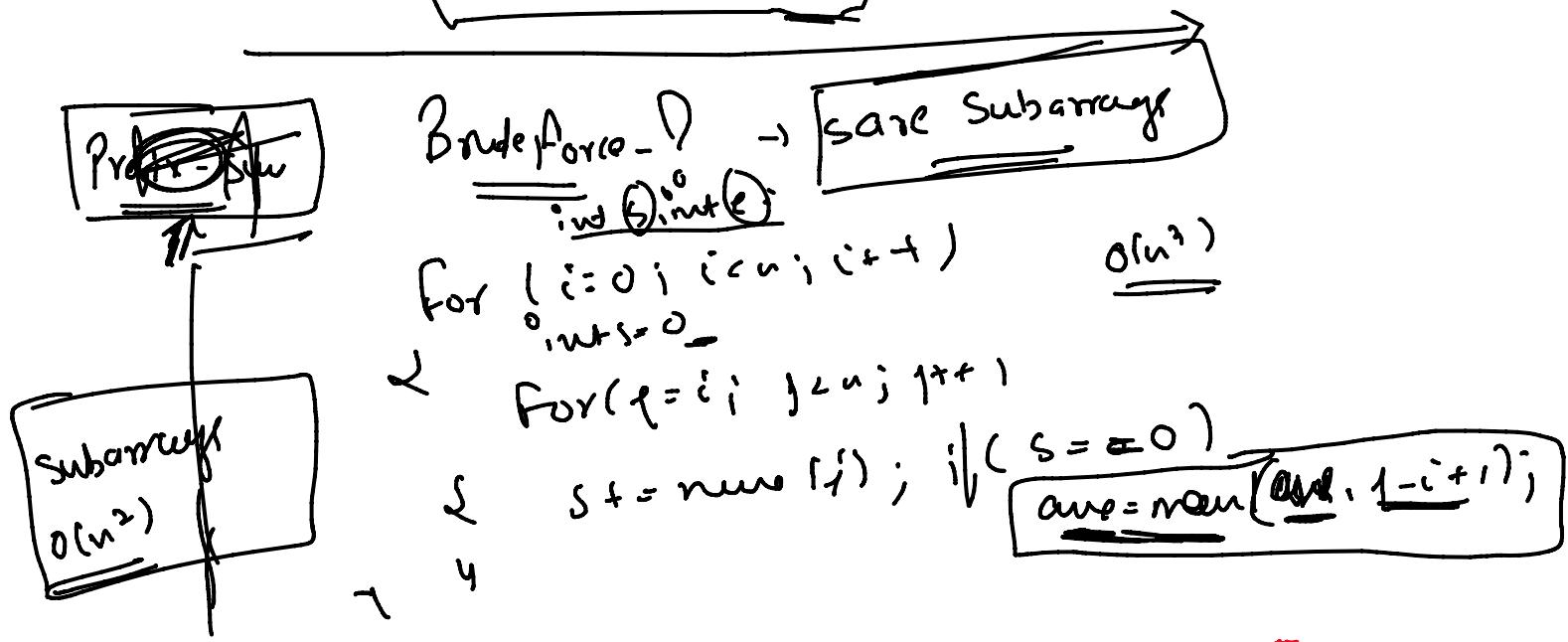
Two Sum

else if $(a[i] + b[j]) = \text{target}$

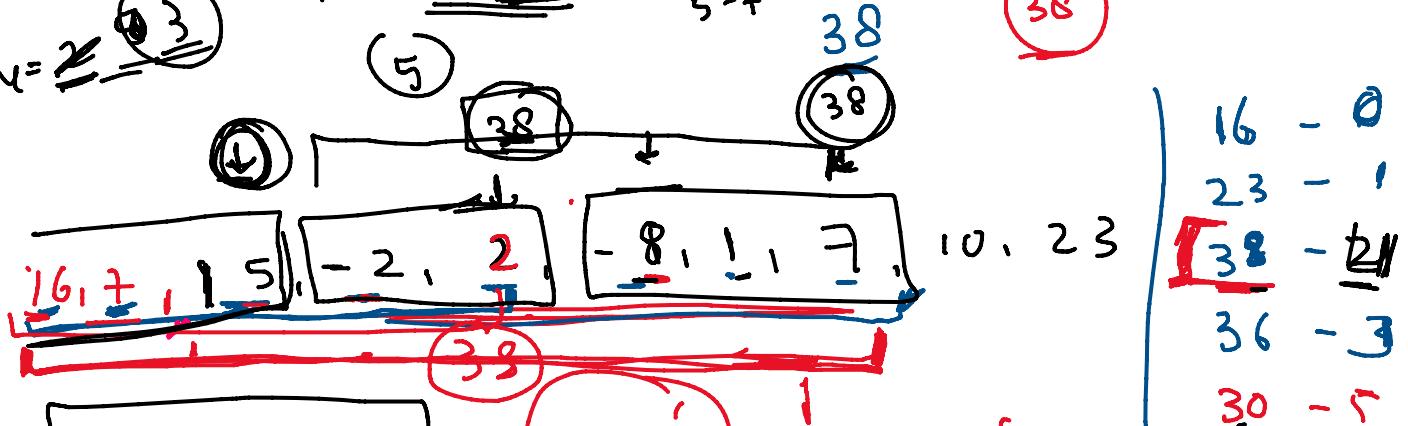
\downarrow
return;

target

15, [-2, 2, -8, 1, 7], 10, 23



~~Point ($s - e$)~~ 5 → 38 ↑ index
~~ans = 2~~ 38



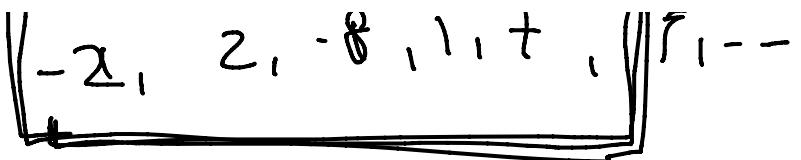
$$\text{Pre}[j] - \text{Pre}[i] = i+1-j = \text{Subarray Size}$$

$$\text{Subarray} \Rightarrow \text{sum} = 0 \quad \text{Pre}[j] - \text{Pre}[i] = 0$$

$$\text{Pre}[i] = \text{Pre}[i] \quad (i+1-j) \quad \text{sum} = 0$$

$$\text{Pre}[j] = i+1-j = 0 \quad [-2, 2, -8, 1, 7, 10, 23]$$

Pre[i]



Vector<int> Pre[n];

Pre[0] = 0

Pre[0] = num[0];

map<int, int> m;

m[num[0]] = 0;

for (i=1; i < n; i++)

↓ Pre[i] = Pre[i-1] + arr[i];

if (m.find(Pre[i]) != m.end())

ans = max(ans, i - m[Pre[i]]);

~~if~~ maxPre[i] = 1

y