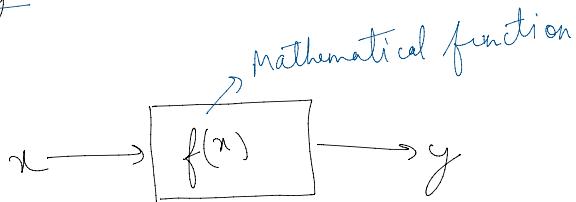
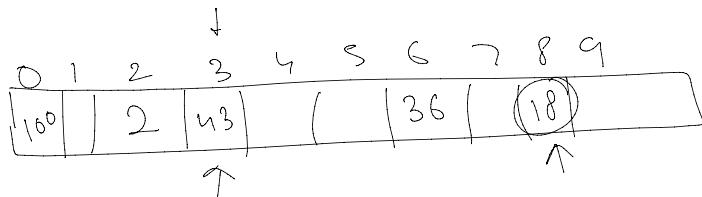


Hashing

$$\Rightarrow [2, 18, 36, 43, 100] \xrightarrow{x \rightarrow \boxed{x \% 10}} y$$

$\Theta(1)$



$O(n)$  Time complexity

$$\textcircled{18} \% 10 = 8$$

$$\textcircled{33} \% 10 \Rightarrow \textcircled{3}$$

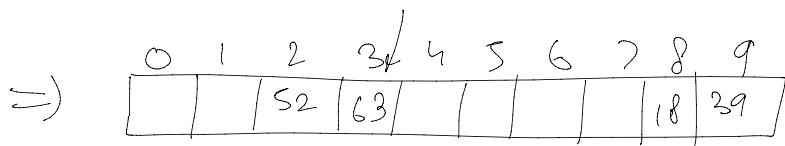
Searching:  $\Theta(1)$

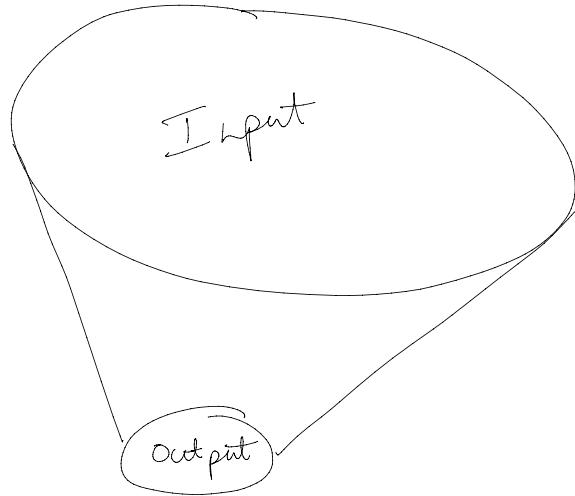
1) Collision

$$x \rightarrow \boxed{x \% 10} \rightarrow y$$

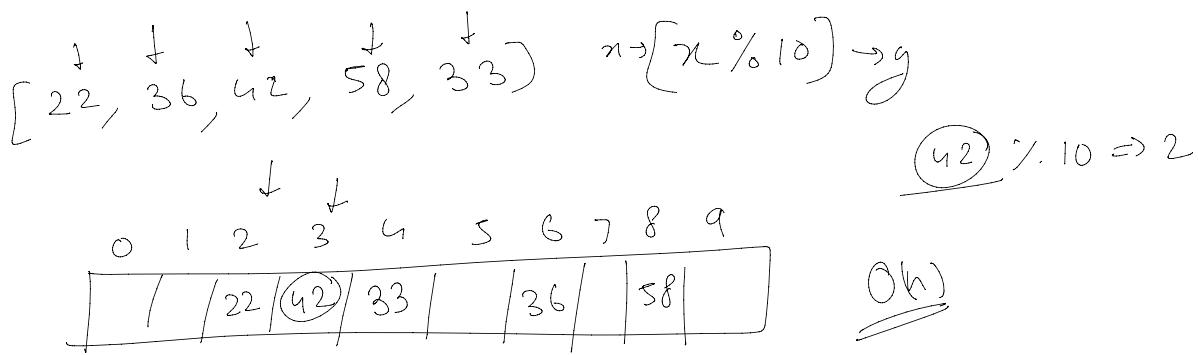
When more than one number map to the same value.

$$[52, 39, 63, 18, 43, 56, 98]$$



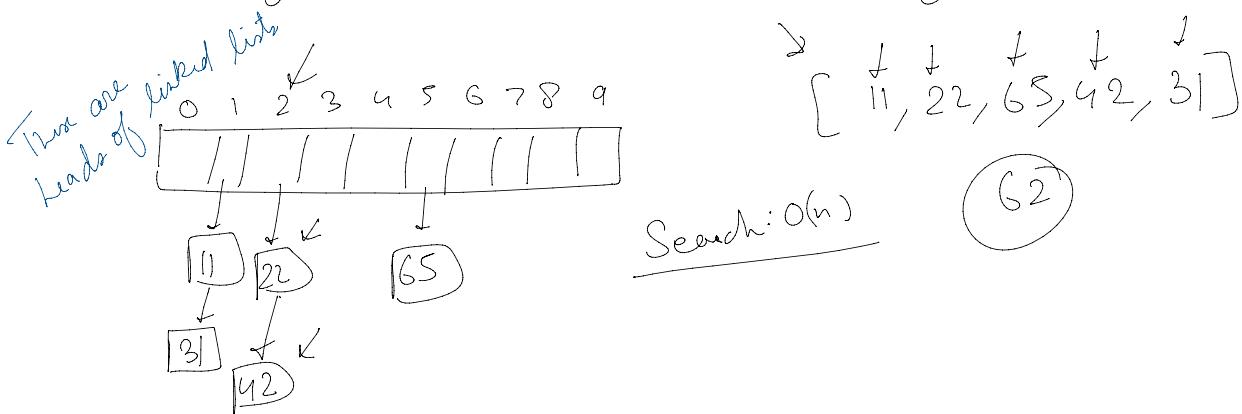


### 1) Linear Probing



### 2) Chaining

$$x \rightarrow [x \% 10] \rightarrow y$$



### Subarray with 0 sum

$$[4, \underline{2, -3, 1}, 6]$$

$\underline{2 + -3 + 1 = 0}$

↓

true

I starting index.

① for (int i=0; i < A.size(); i++) {

② for (int j=i; j < A.size(); j++) {

int sum = 0;

③ for (int k=i; k < j; k++) {

sum += A[k];

}

if (sum == 0) return true;

}

return false;

$\mathcal{O}(n^3)$

[4, 2, -3, 1, 6]

[4, 6, 3, 9, 10] → 0

Prefix Sum

[5, -2, -3, 8]

[5, 3, 0, 8]

1) Initialize a map and a variable sum = 0.

[ $\frac{1}{5}, \frac{9}{4}, \frac{3}{-2}, \frac{-1}{-1}, \frac{6}{6}$ ]

2) In the map, put the value 0.

Sum = 0

3) Iterate the array(i):

a) sum += A[i]

b) If map already has value sum stored in it, return true.

c) store sum into the map.

4) return false

[0:1  
5:1  
9:1  
12:1  
10:1]

[1, 10, 93, 3, 5]

[0:1  
1:1  
2:1  
3:1  
..]

Sum = 0 + 1 + 10 + 93 + 3 + 5

$$\text{Sum} = 67 + 710 + 113 + 118$$

Time:  $O(n)$

Space:  $O(n)$

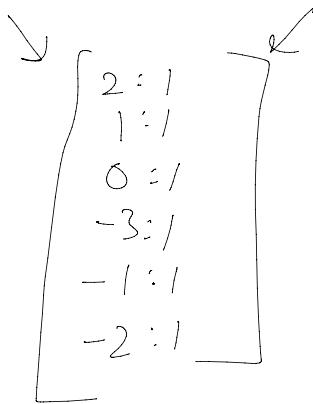
17:1
110:1
113:1
118:0

### Consecutive array elements

$$\Rightarrow [2, 1, 0, -3, -1, -2]$$

$$\min_{\text{val}} = 2 \times \cancel{-3}$$

$$\begin{aligned} & -3 \text{ to } (-3+n-1) \\ & (-3+6-1) \\ & \underline{\underline{(2)}} \end{aligned}$$



T:  $O(n)$       S:  $O(n)$

1) Create a map and a variable to store minimum value.

2) Iterate the array ( $i$ ):

- a) Update the value of  $\min_{\text{Val}} = \min(\min_{\text{Val}}, A[i])$ .
- b) Store the value of  $A[i]$  in the map.

3) Iterate from  $\min_{\text{Val}}$  to  $(\min_{\text{Val}} + n - 1)$ :

- a) If any number is missing, return false.

4) return true.

### Subarray sum equals K

$$\text{array} = [1, 2, 3] \quad K = 3$$

```

I ans = 0;
① for (int i=0; i < A.size(); i++) {
    ② for (int j=i; j < A.size(); j++) {
        int sum = 0;
        ③ for (int k=i; k <= j; k++) {
            sum += A[k];
        }
        if (sum == K) ans++;
    }
}
return ans;

```

$[1, 2, 3]$   $\textcircled{k=3}$

$[1, \underline{3}, \underline{6}]$  Prefix Sum

$[2, \underline{3, 6, -2}, 4]$   $\textcircled{k=7}$

$\Rightarrow [2, \underline{5, 11, \textcircled{9}}, 13]$  Prefix Sum

$\Rightarrow [2, \underline{3, 6, -2, 10, -10}, 4]$   $\textcircled{\cancel{k=7}}$

$\underline{[2, 5, 11, \textcircled{9}, 19, \textcircled{9}, 13]}$

... and ...



Ques 1

- 1) Create a map and a variable sum = 0 and ans = 0
- 2) m[0] = 1.
- 3) Iterate the array (i):

- a) sum += A[i]
- b) ans = m[sum - k]
- c) m[sum]++

4) return ans.

T: O(n)

S: O(n)

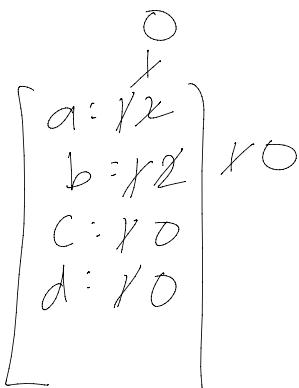
	Sum	ans
	0	0
	1	
	3	
	5	(5)
	11	

0: 1	19: 1
2: 2	13: 1
7: 1	
5: 1	
11: 1	
9: 12	

Anagram

$$\rightarrow s = \begin{matrix} & + & + & + & + \\ a & b & c & d & a & b \end{matrix}$$

$$t = \begin{matrix} & + & + & + & + \\ a & a & b & c & b & d \end{matrix}$$



- 1) Create a map.

T: O(n)      S: O(1)

- 2) Iterate strings:

a) store count of each letter into the map.

- 3) Iterate string t:

a) delete the count of each from the map.

i) If frequency of each letter == 0, return true.

5) return false.

Largest substring without repeating characters

$\checkmark$   $s = \frac{\text{abc abc bb}^{\boxed{b}}}{\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow}$

~~abc~~  
~~abc~~

b ca  
c ab

$$\left[ \begin{array}{r} a : 3 \\ \hline b : 6 \\ \hline c : 5 \end{array} \right]$$