

Hashing

	key	value		value
Room no.	Avail.	Brock	cost	
101	X			
102	✓			
103	✓			
:				

index \Rightarrow room no

$$\text{arr[1]} \Rightarrow \text{arr[\underline{\text{room no}}];} \quad \underline{\text{arr[02]}}$$

$1 \rightarrow 10^6 \Rightarrow \text{easy}$

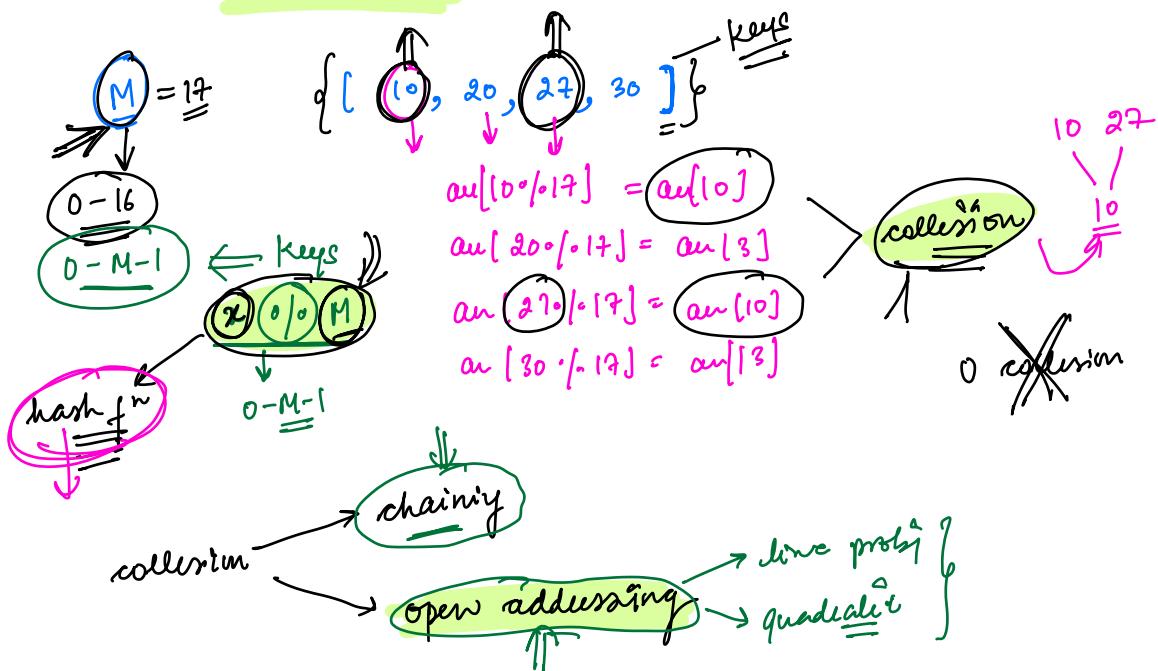
$\approx 10^6 \text{ memory}$

$\underline{14567892}$

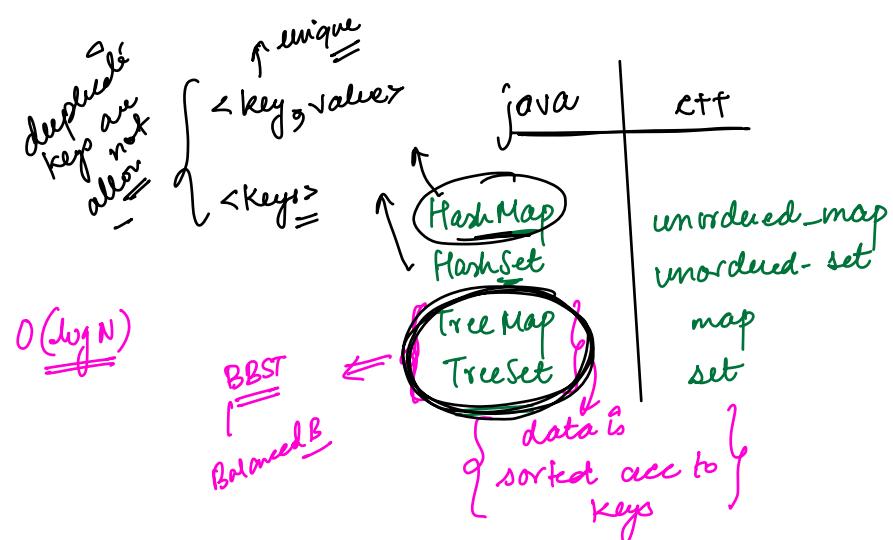
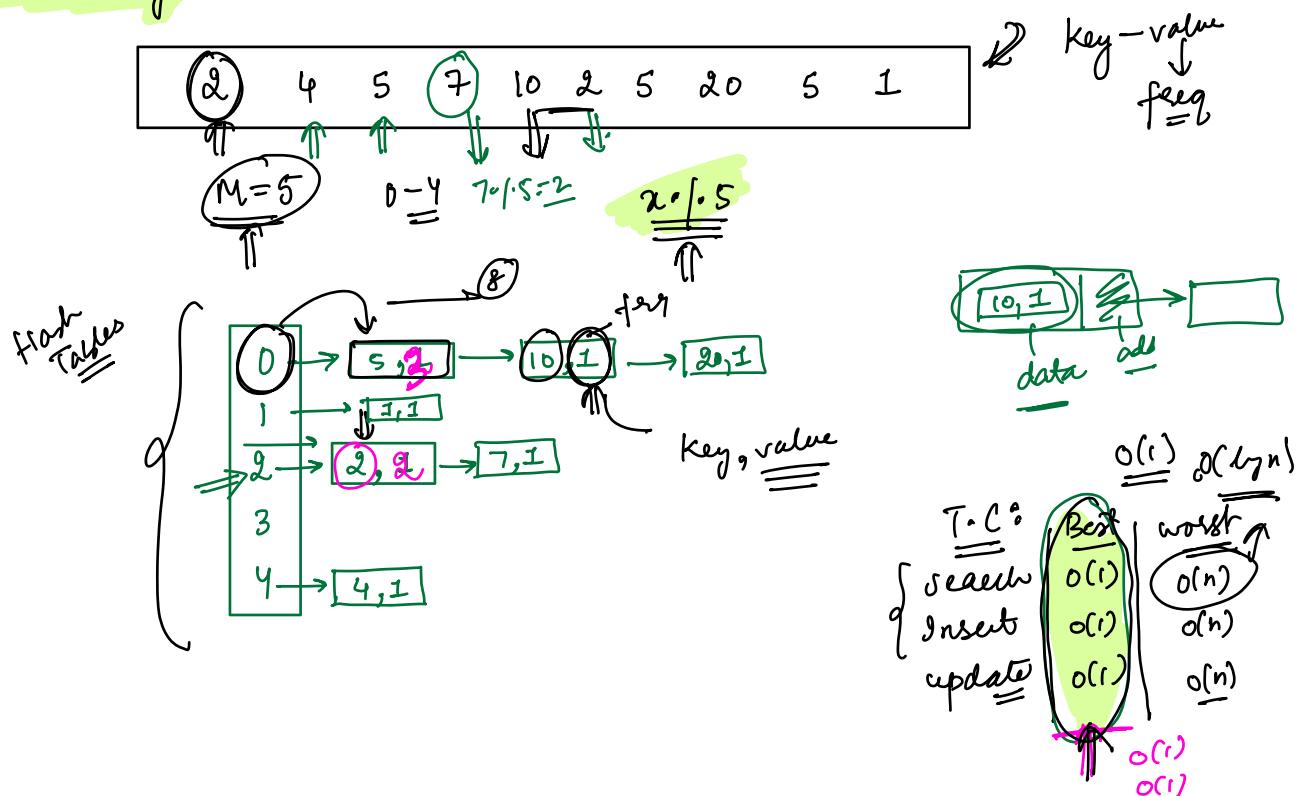
access - $O(1)$
update - $O(\underline{1})$

$$\begin{array}{c} \text{min} \quad \text{max} \\ \downarrow \quad \downarrow \\ 0 \quad 1 \quad \dots \\ \hline \text{range} = 10^6 \end{array}$$

$$1 \leq A[i] \leq 10^9$$



Chaining

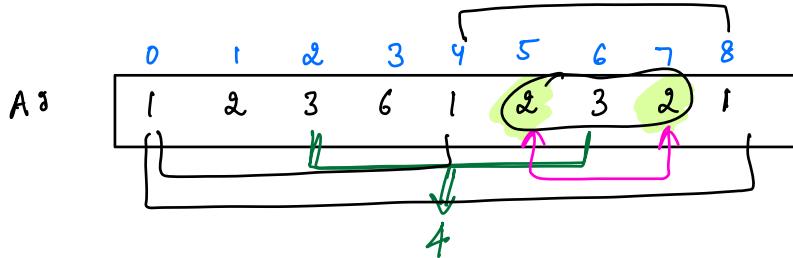


~~Set.insert(s)~~
~~Set.insert(s)~~
java-overview

Q Given an integer array, find any pair (i, j) such that

$$\underline{j > i} \quad A[i] == A[j] \text{ & } (j-i) \text{ is minimum.}$$

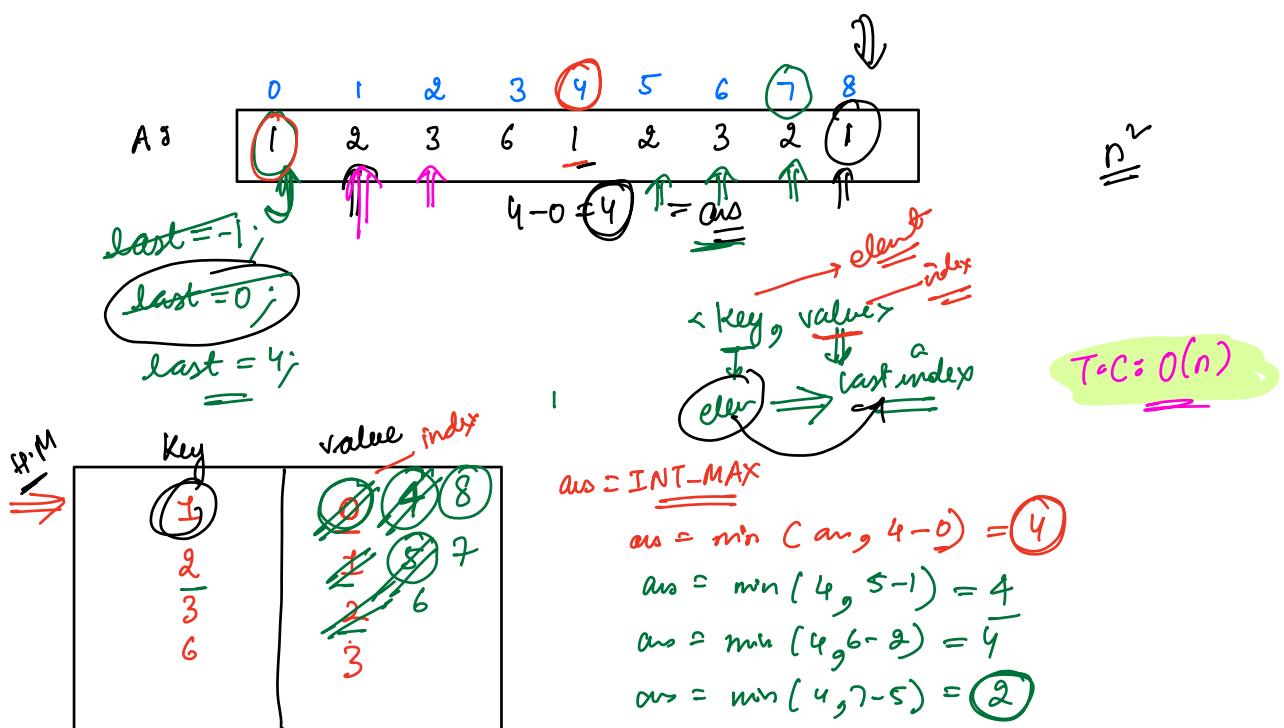
\downarrow
closest duplicate pair



B.F.E consider all pairs $\Rightarrow O(n^2)$

| - - - | - - - | . . . | . . .

any alert \Rightarrow when did you last find something same as me?



HM
for $i = 0 \rightarrow n$

if ($an[i]$ present in HM)

{ $ans = \min(ans, i - HM(an[i]))$;
 $HM[an[i]] = i$;

}

else

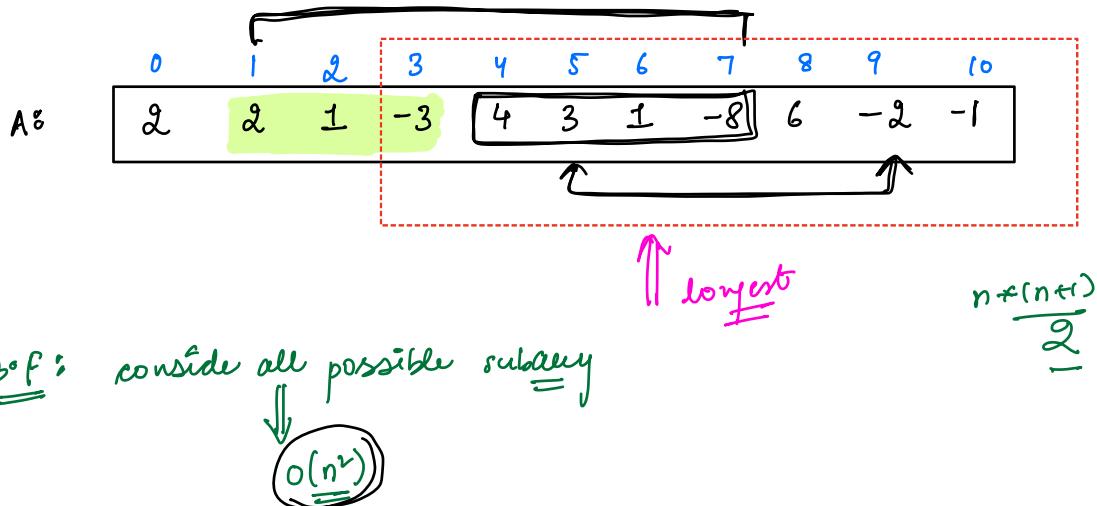
{ $HM.insert(an[i], i)$;

}

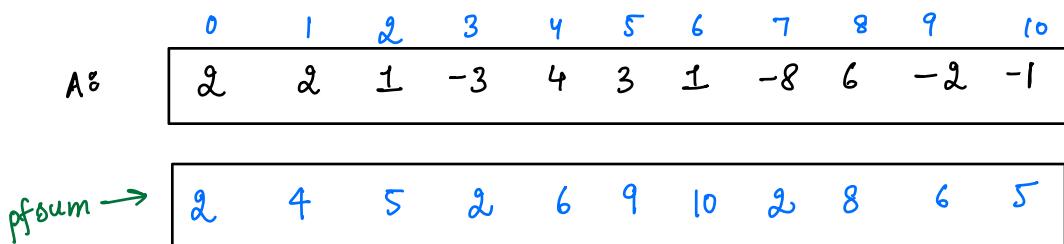
}

hm['key'] \rightarrow value

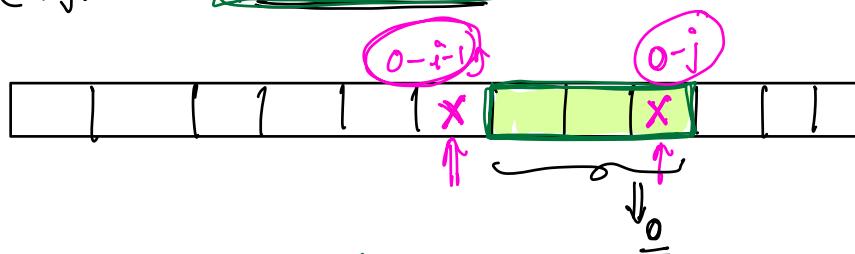
Q2 Given an integer array. Find longest subarray with sum = 0.



prefix sum



$$(i, j) \Rightarrow \underline{\underline{pf[j] - pf[i-1] = 0}}$$

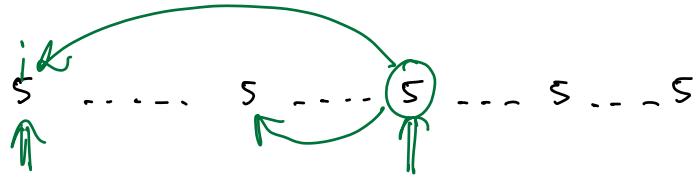


$$pf[j] - pf[i-1] = 0$$

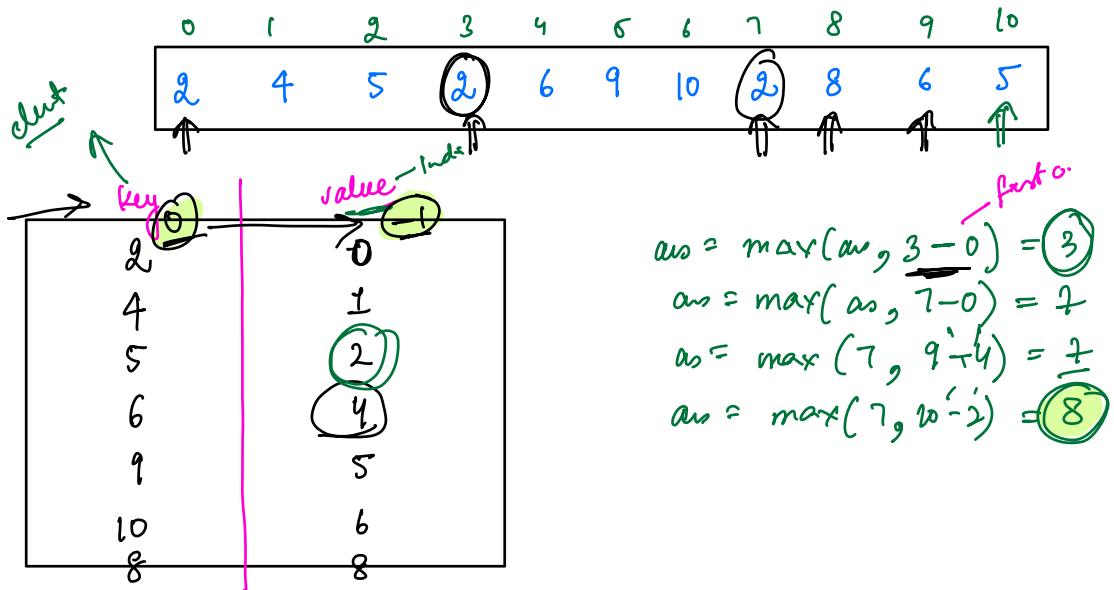
$$\Rightarrow pf[j] = pf[i-1]$$

farthest duplicates

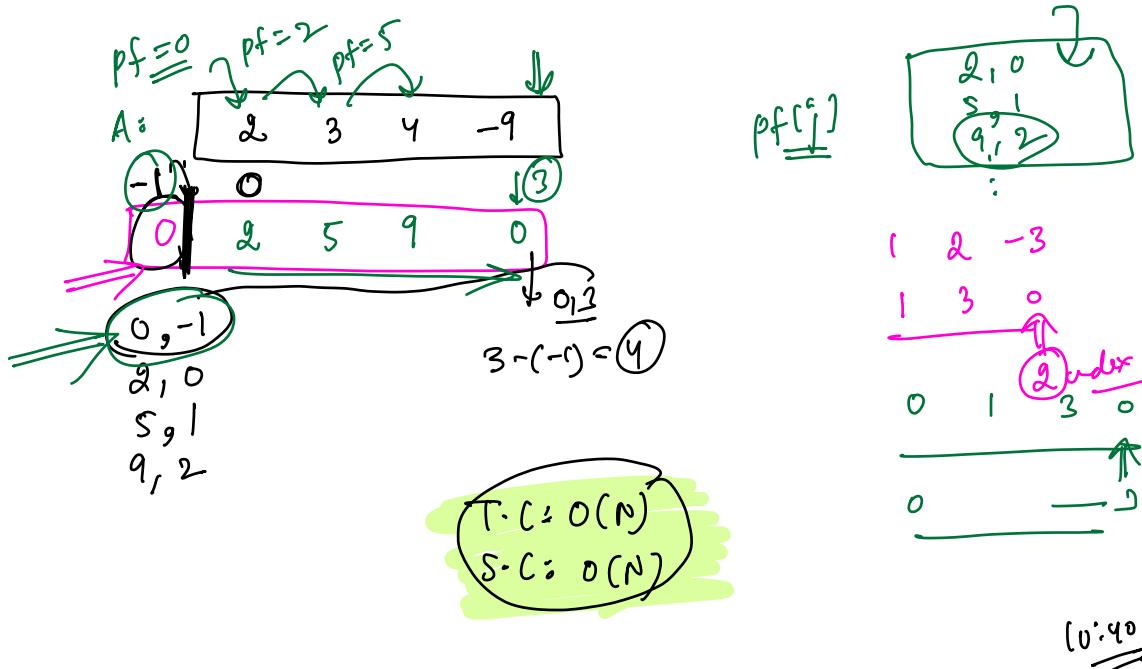
we need to look for duplicate in pf's



Every element wants to know its first occur.

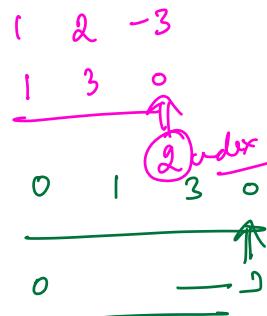
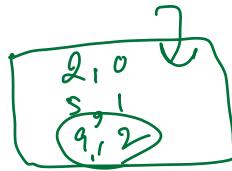


$$\begin{aligned}
 \text{ans} &= \max(\text{ans}, 3 - 0) = 3 \\
 \text{ans} &= \max(\text{ans}, 7 - 0) = 7 \\
 \text{ans} &= \max(7, 9 - 4) = 7 \\
 \text{ans} &= \max(7, 10 - 2) = 8
 \end{aligned}$$



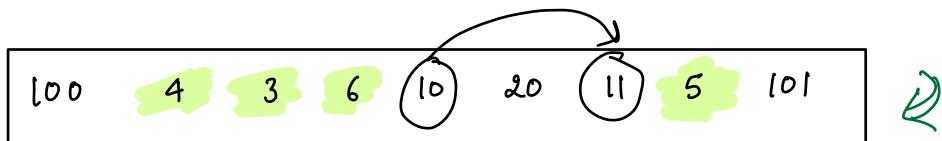
$T.C = O(N)$
 $S.C = O(N)$

$pf[0]$



$O(N)$

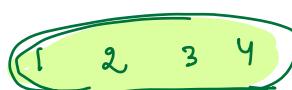
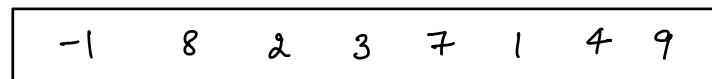
Q Given an array. Find the length of largest sequence which can be rearranged to get consecutive elements.



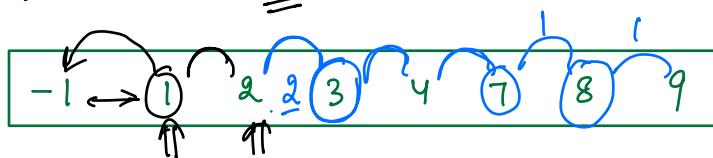
$$10 \rightarrow 11 \equiv 2$$

$$100 \rightarrow 101 \equiv 2$$

$$3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \equiv 4$$

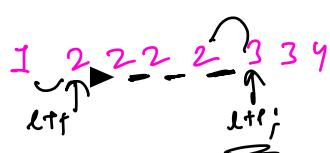


① sort the data



$l=1$ $l+p$ $l+p$ $l+p$ $ans = \text{array}[ans, l]$
 $\underline{l=2}$ $\underline{l=3}$ $\underline{l=4}$ $\underline{l=1}$ $l \leftarrow l+1$
 || sort $chain = 1 / ans = 0$ $i = 1 \rightarrow i < n$

T.C: $n \log n$



$\text{if } (\text{arr}[i] == \text{arr}[i-1] + 1)$

$\quad \quad \quad \text{chain} += 1;$

$\text{else if } (\text{arr}[i] == \text{arr}[i-1])$

$\quad \quad \quad ans = \max(ans, chain); \quad ans = 0;$

$\quad \quad \quad chain = 1;$

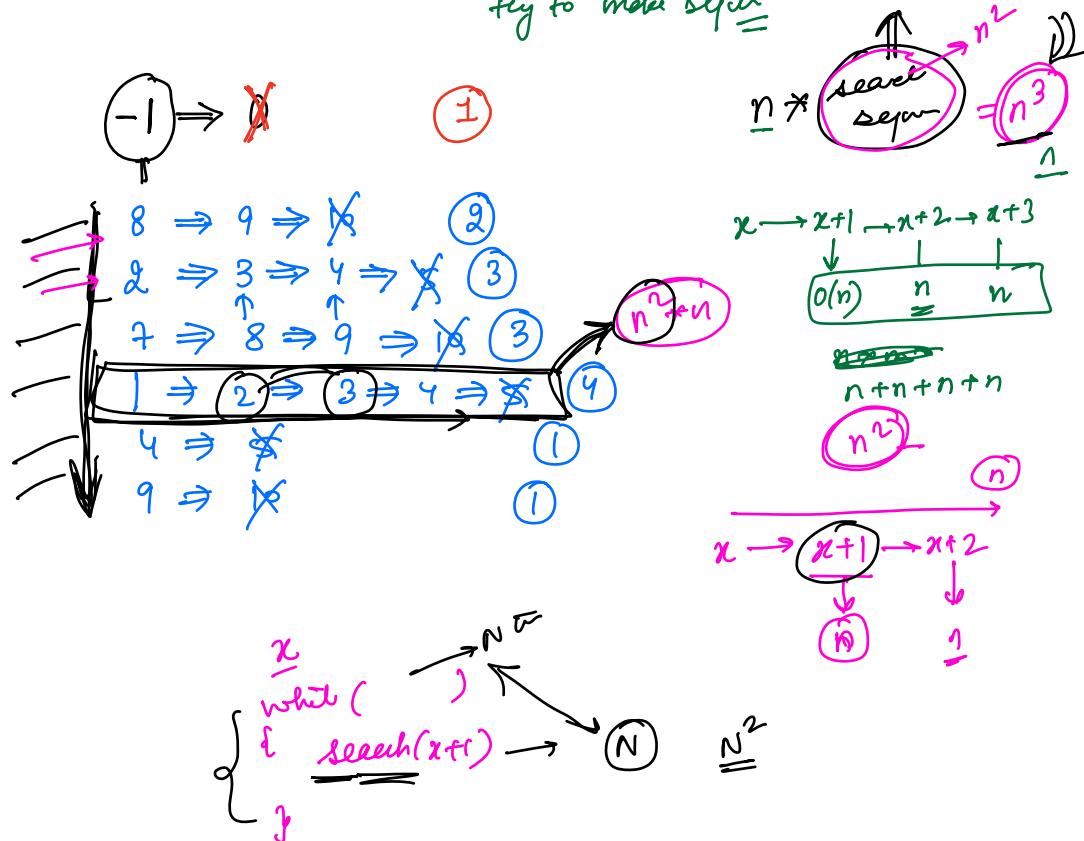
$\Rightarrow ans = \max(ans, chain);$

1 2 3 8 9 10 11 12 13 14

-1	8	2	3	7	1	4	9
↑		↑					

B.F

consider every element as your start point & try to make sequ $\underline{\underline{=}}$



Set

-1	8	2	3	7	1	4	9
----	---	---	---	---	---	---	---

\Leftrightarrow ~~set~~ $- O(\underline{\underline{n}})$

-1	8	2	3	7	1	4	9
----	---	---	---	---	---	---	---

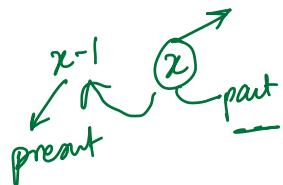
with set = search for an element $- O(\underline{\underline{1}})$

building a sequ $- O(n)$

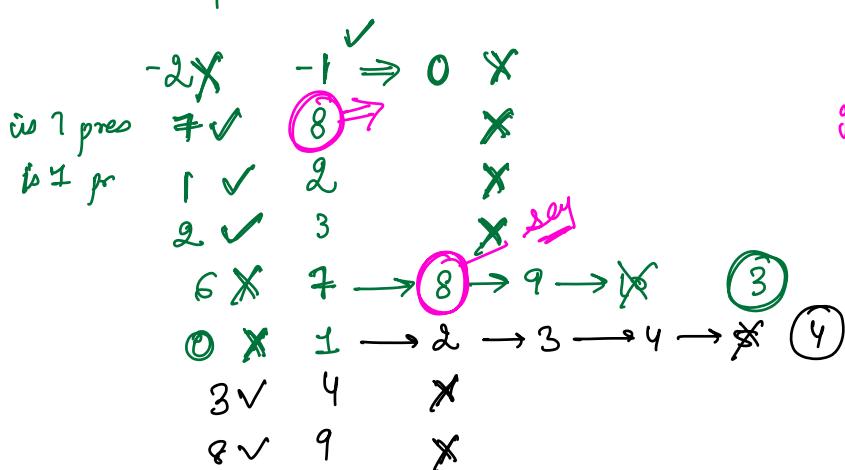
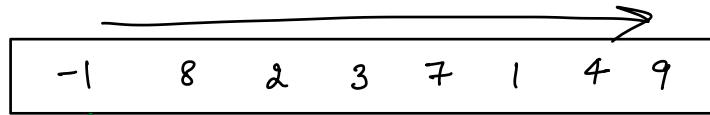
T.C: $O(N^2)$

N^2
 \downarrow
 N^2
 \downarrow
 N

How will you know if a particular element
will start a sequence?



If $x-1$ is present,
 x is not the start



$$T.C. = 2N + N \underset{\approx}{=} O(N)$$

I will only
search for an
element twice
one while
creating the
sequence,
other as
a start

presence → key → set

// build the hash set

```

for( i=0 → n)
{
    if( arr[i]-1 is not present)
    {
        chainLength = 1;
        x = arr[i]+1;
        while( x is present)
        {
            chainLength++;
            x++;
        }
        ans = max(ans, chainLength);
    }
}

```

→ ans arr[i]-1

→ second gr.

count of distinct element → set ? → set.size()

