

CLASS 2.6

Stack      S-Q

Queues using Stack → 2

Approaches →  $O(n)$

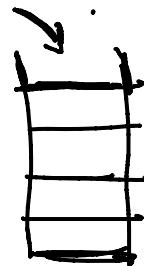
- Push costly →  $O(n)$
- Pop costly →  $O(n)$
- Top →  $O(n)$
- Print(d) →  $O(1)$

Stack using array

→ min 2 arrays

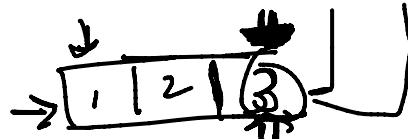
→ Push costly

→ Pop costly



top element

last inserted = element



firstly inserted

array < n > A[ ]

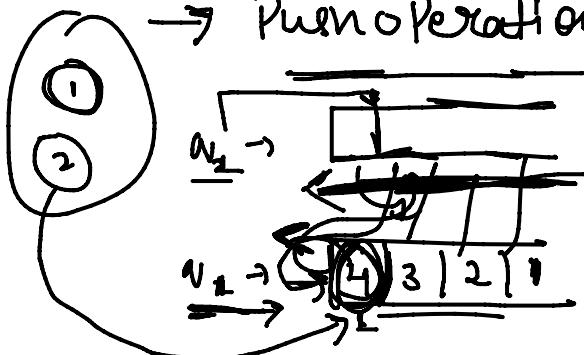
n = rear( ) X

array =

1 array



→ Push operation costly



min 2 arrays



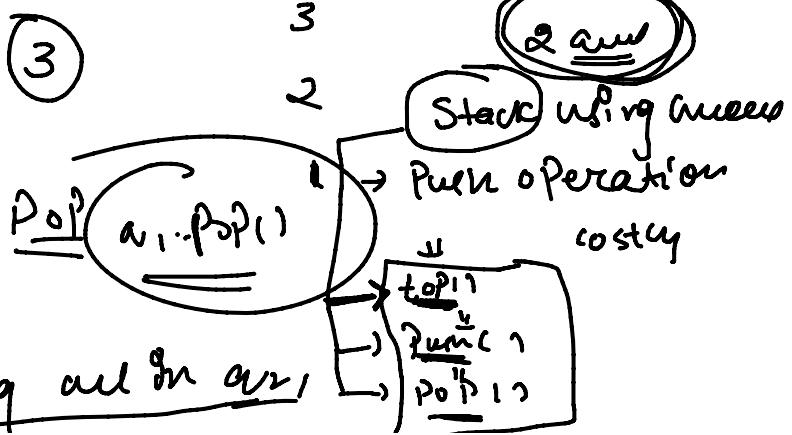
(3)

4 top(1) → arr1.front(1)

Push

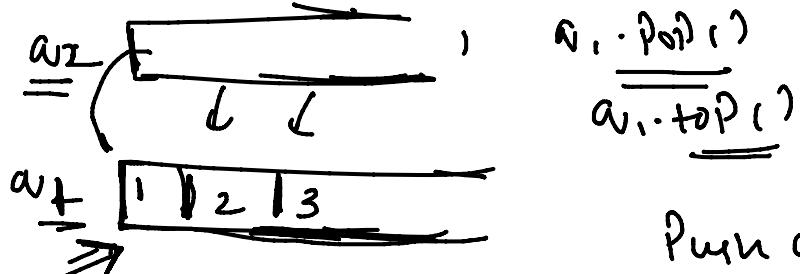
{ Push(in arr2)

= Empty arr1 and Pushing all in arr2

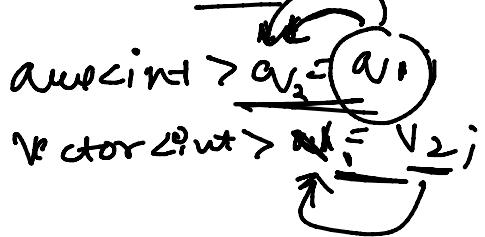


\* push in  $a_2$   
 \* Empty  $a_1$  and Pushing all in  $a_2$ ,  $\xrightarrow{\text{sum } 7}$   
 \* Reverse (swap  $a_1, a_2$ )

Pop operation costly



Swap ( $a_1, a_2$ )



Push operation O(1)

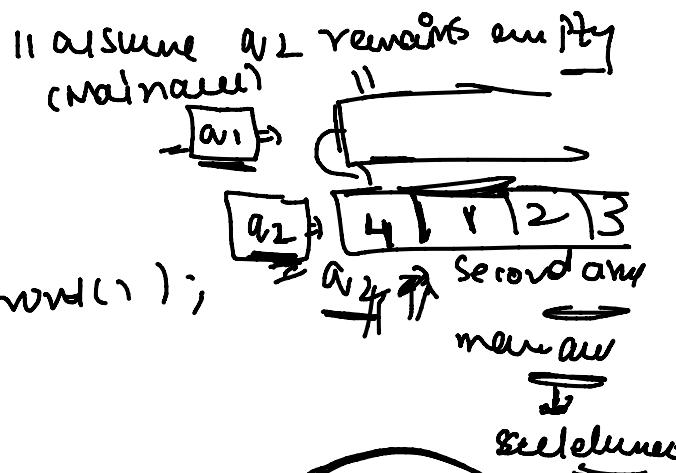


Push operation costly

move cost >  $a_1, a_2$

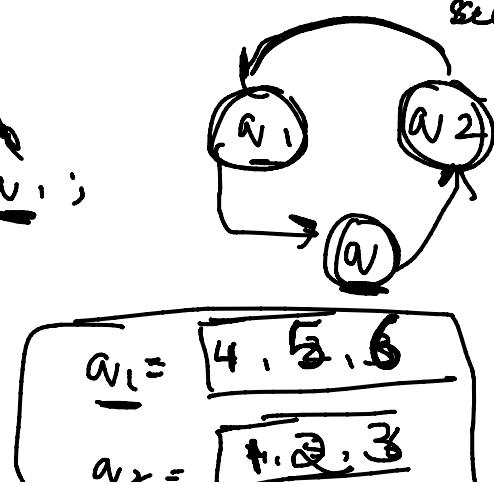
```

void Push(int u)
{
   $a_2 \cdot \text{push}(u)$ ;
  while (! $a_1 \cdot \text{empty}()$ )
  {
     $a_2 \cdot \text{push}(a_1 \cdot \text{front}())$ ;
     $a_1 \cdot \text{pop}()$ ;
  }
}
  
```



swap both arrays

swap ( $a_1, a_2$ )  
 $\xrightarrow{\text{move cost } > a_1 = a_2;}$   
 $a_1 = a_2;$   
 $a_2 = a_1;$



γ

Void pop()

{ if (a<sub>1</sub>.empty())  
 ↳ cout << "Queue is Empty";

a<sub>1</sub>.pop(); O(1)

γ

~~int top()~~

{ if (a<sub>1</sub>.empty())  
 ↳ return -1;  
 return a<sub>1</sub>.front(); O(1)

γ

2<sup>nd</sup> Approach

Make Pop costly

queue<int> a<sub>1</sub>, a<sub>2</sub>;

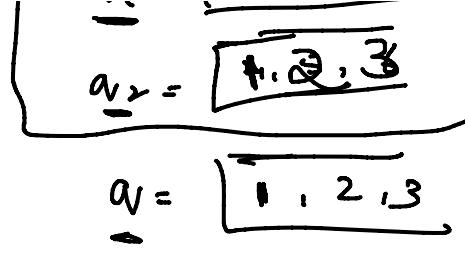
Void pop()

{ if (a<sub>1</sub>.empty())  
 ↳ cout << "Empty";  
 γ

while ( a<sub>1</sub>.size() != 1 )

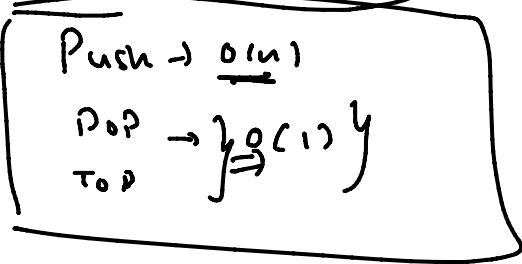
{ a<sub>2</sub>.push(a<sub>1</sub>.front())  
a<sub>1</sub>.pop();

γ



Push → O(n)

Pop → O(1)



void top()

{ if (a<sub>1</sub>.empty())  
 ↳ return -1;

while ( a<sub>1</sub>.size() != 1 )

{ a<sub>2</sub>.push(a<sub>1</sub>.front())

γ

int = a<sub>1</sub>.top();

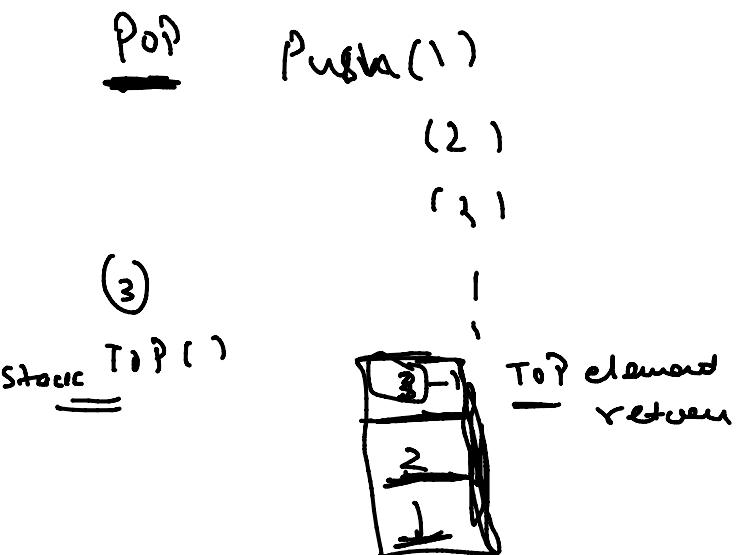
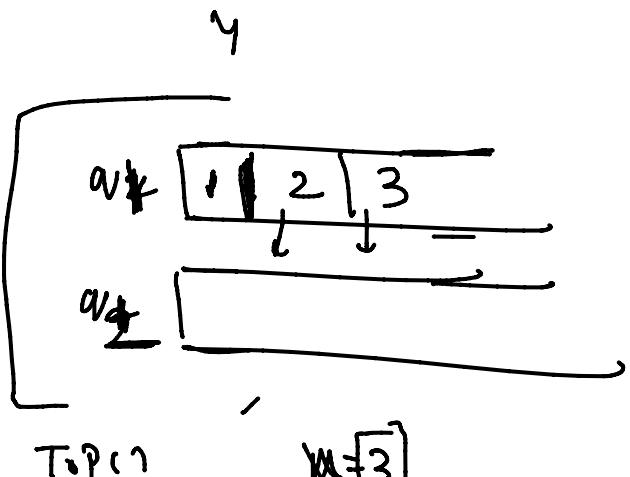
a<sub>1</sub>.pop();

γ

$\text{a}_1 \cdot \underline{\text{pop}();}$   
 $\underline{\text{swap}(\text{a}_1, \text{a}_2);}$   
 $\text{a}_2$

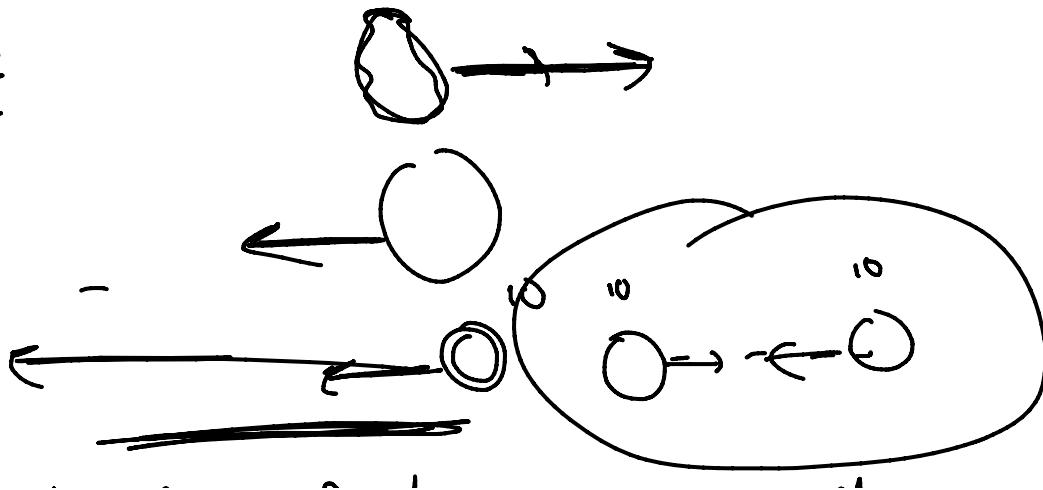
$\text{a}_1 = \underline{\text{a}_2}$   
 $\text{a}_1 \cdot \text{pop}();$   
 $\underline{\text{a}_2 \cdot \text{push}(\underline{\text{a}_1});}$   
 $\underline{\text{swap}(\text{a}_1, \text{a}_2);}$   
 $\underline{\text{getvalue};}$

void Push (int u)  
 $\{ \text{a}_1 \cdot \text{push} (\text{u});$

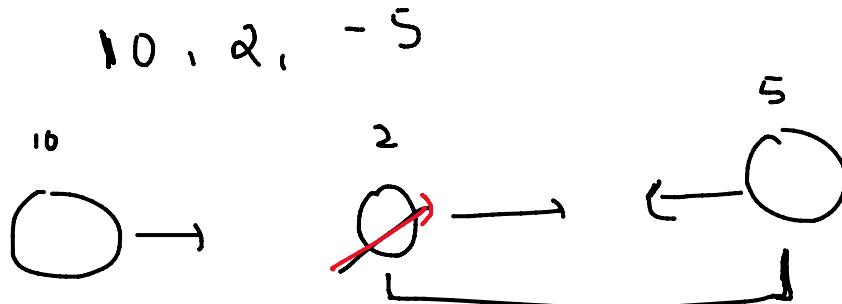
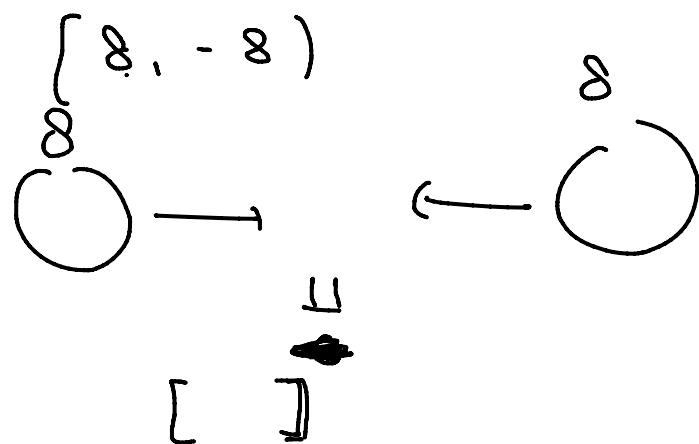
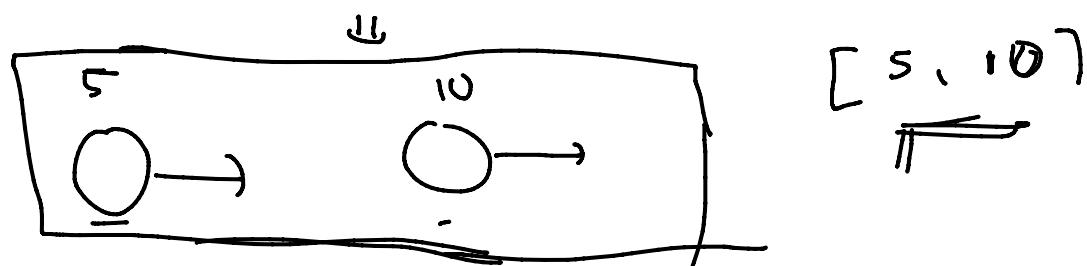
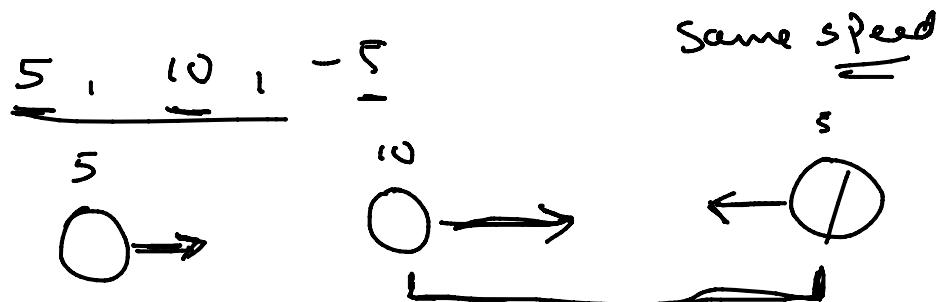


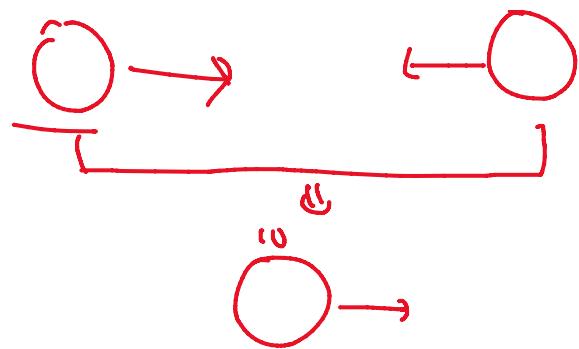
Storage      Direct      store  
Asteroid collision + 2

$+ (5, 10, -5)$

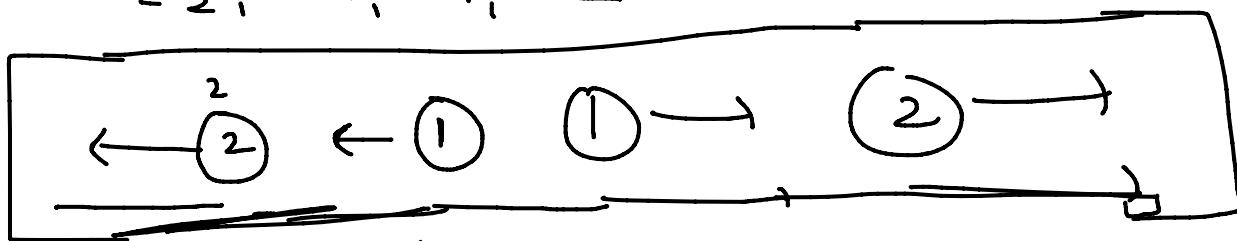


- ① smaller size explode  
 ② Both equal Both explode

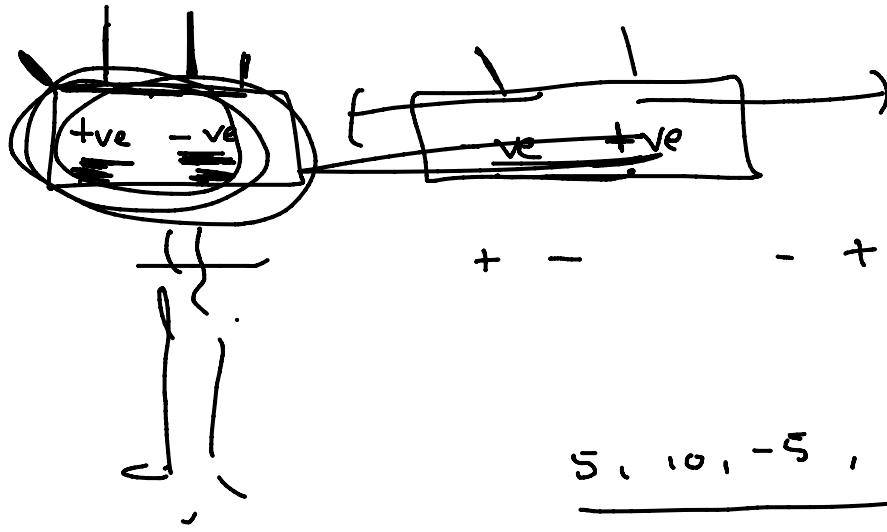




-2, -1, 1, 2

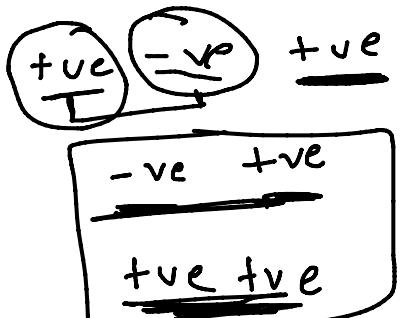


final



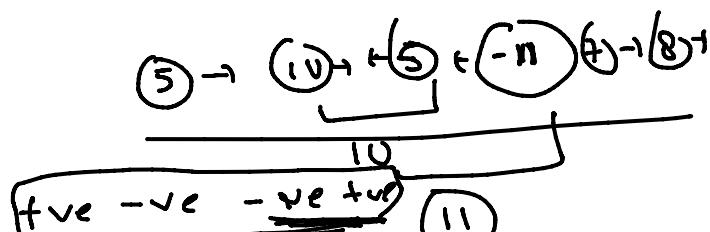
Stack (ind) Positive ;

Stack (ind) negative ;



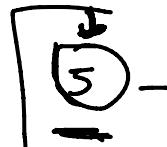
5, 10, -5, -11, 7, 8

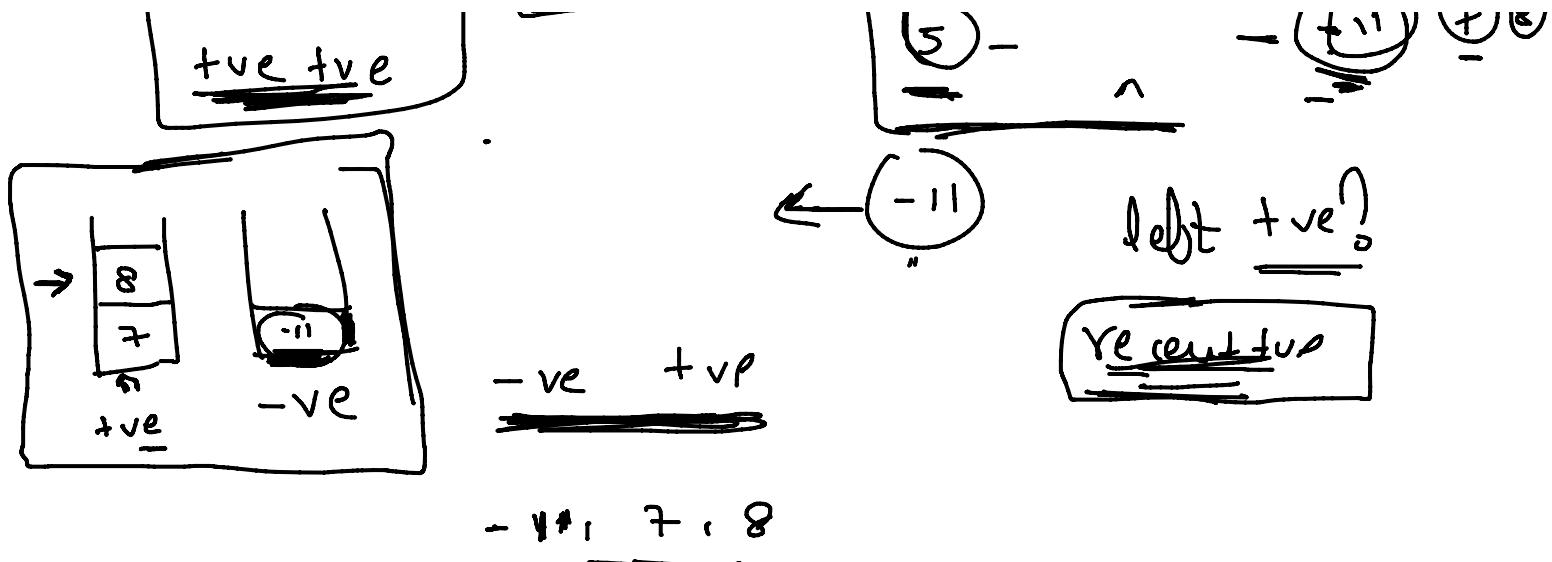
-11 7 8



5 → ← (11) 7 → 8

← (11) 7 → 8 →





Slack < 0 > negative, Positive

For (int Asteroid = 0; Asteroid < Asteroids.size(); Asteroid++)

2

if (a[i] > 0)  
    & Positive. Push(i);

3

+ve    -ve  
else  
    bool flag = 1  
    while (! Positive.empty())  
        if (a[Positive.top()] < abs(a[i]))  
            Positive.pop();  
        else if (a[Positive.top()] = -abs(a[i]))  
            Positive.pop();  
        flag = 0, break;  
    if (flag = 0) ...

1) flag = 0

L1 a[i] > 0 ...

L, ali(i) exploded . ?

+ flag = 0, break)

else —

{ flag = 0; asteroid exploded  
    break;

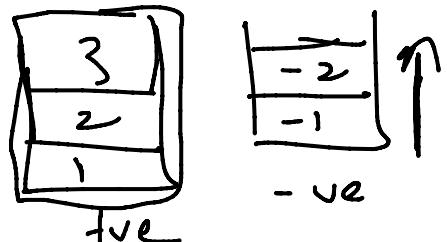
?

if (flag = 1) & negative . Push(i);

? ?

Vector<int> ans;

-ve +ve



while (!negative . empty(1))

{ ans.push-back(negative . top(1));  
    negative . pop();

?

reverse ( ans.begin(), ans.end() );

int s2 = ans.size();

while (!Positive . empty(1))

{ ans.push-back(Positive . top());

    Positive . pop();

?

reverse ( ans.begin(), +s2, ans.end() );

↓ ↓

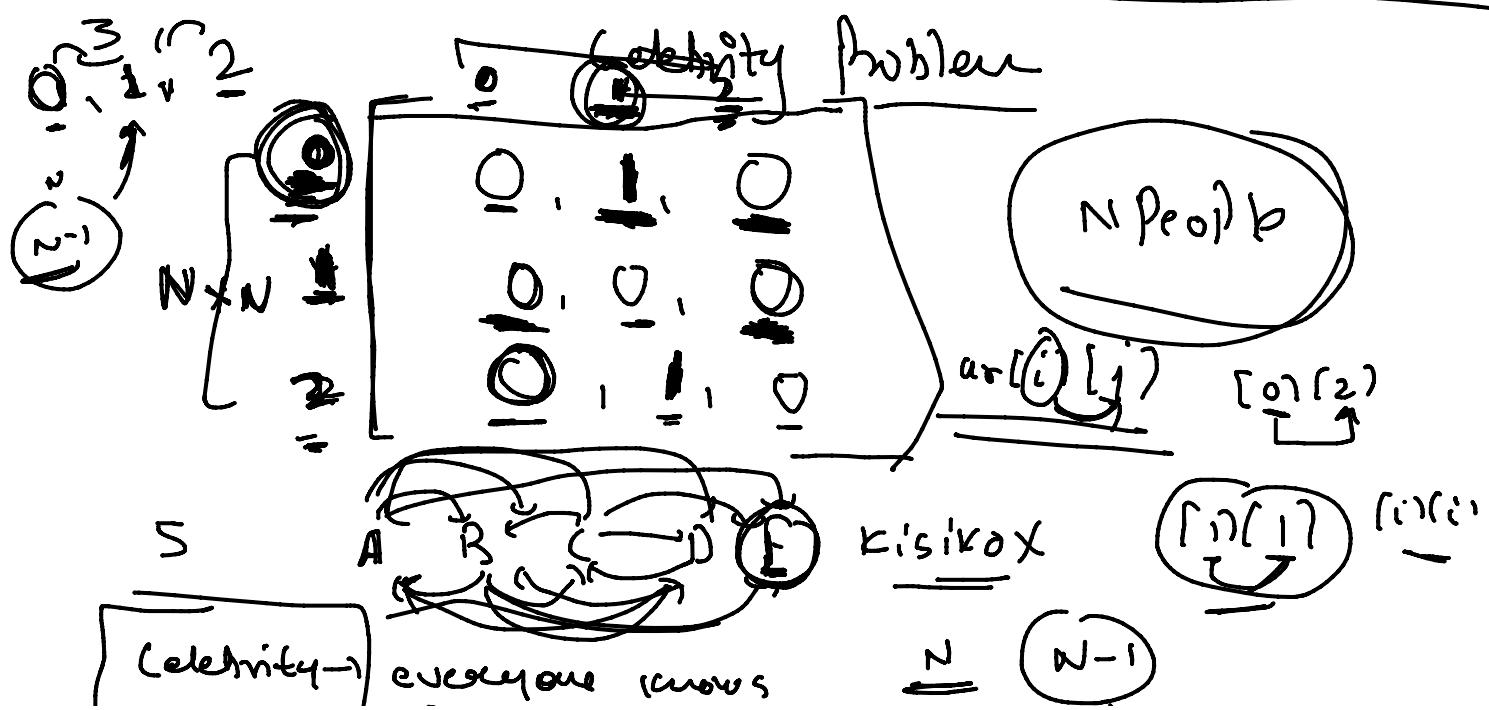
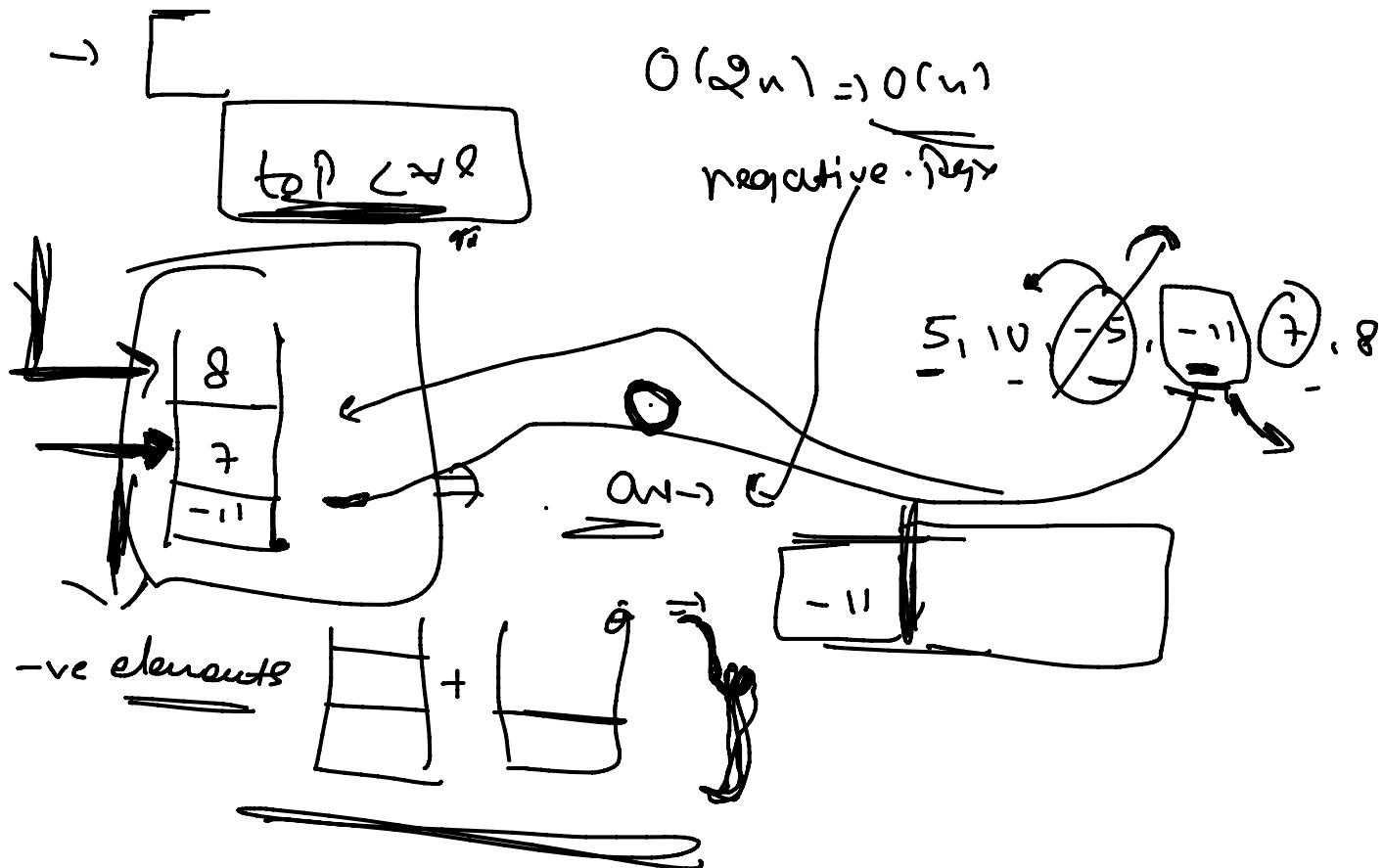
↓ ↓

(2)

-1, -2, 1, 2, 3

$\text{ans}[s-2]$

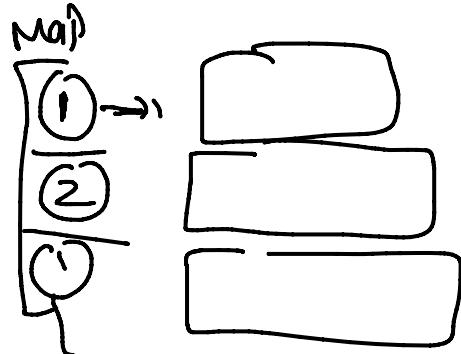
$\text{cur}(n-1)$



Celebrity  $\rightarrow$  everyone knows  $\xleftarrow{N}$   $N-1$   
 $\rightarrow$  They don't know anyone

Brute force  $\rightarrow$  Record  $\rightarrow$   $N-1$

$N^2$



Optimized Using Stack

~~1, 2, 3, 4~~

=

~~(2)(3)=1~~  $\star$  (celebrity?)

[3][1] 3 celebrity?

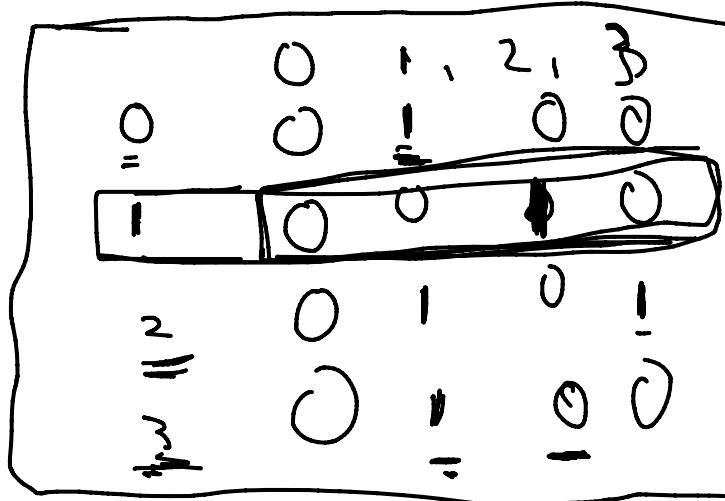
1 chance

~~Can be celebrity~~

$(1, 0) [0][1] = 1$

0 celebrity?

1 chance  $\frac{1}{2}$



$\underline{N \text{ hi}}$

$\underline{[1][0]}$

1 element

Stack  $\rightarrow$  Top 2 elements

eliminate celebrity  $\times$

flag = 1;

For ( $i=0$ ;  $i < n$ ;  $i++$ )

if (M[s.top() > i])

S.top()

(1,2)

