Stack and Queues

Important Questions

1. Reverse a stack .
2. @uene
3.
5. Stock span Problem
4. Balanced Parenthesis

Array

(i) Size fix

(ii) Accessing O(1)

(iii) Traverse ⇄

(iv) Insertion O(n) ✗

insert & delete

5 6 10 12 15

0th index → O(n)
last pos → O(1)
any pos → O(n)

Linked list

(i) flexible size

(ii) Accessing O(n)

(iii) Traversal →

(iv) Insertion O(n) ✓

0th pos → O(1)
last pos → O(n)
any pos → O(n)

STACK

↳ abstract data type

(specific order me data insert/delete karna hota hai)

LIFO

↓

last in first out

eg. push (10)
(20)
(30)

pop //30
pop //20
pop //10

(I) push()
(II) pop()
(III) isEmpty()
(IV) top()
(V) size()

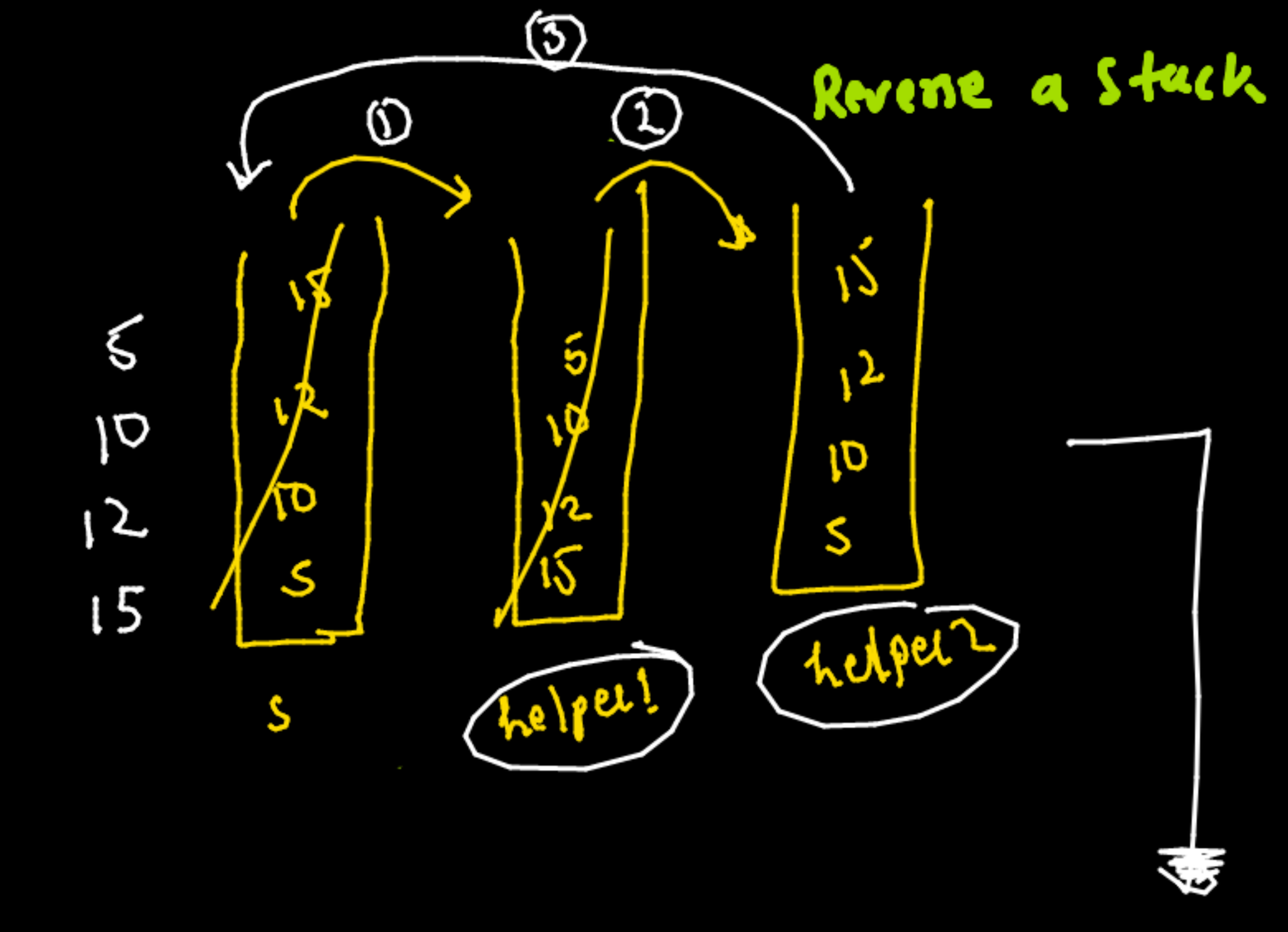
Template

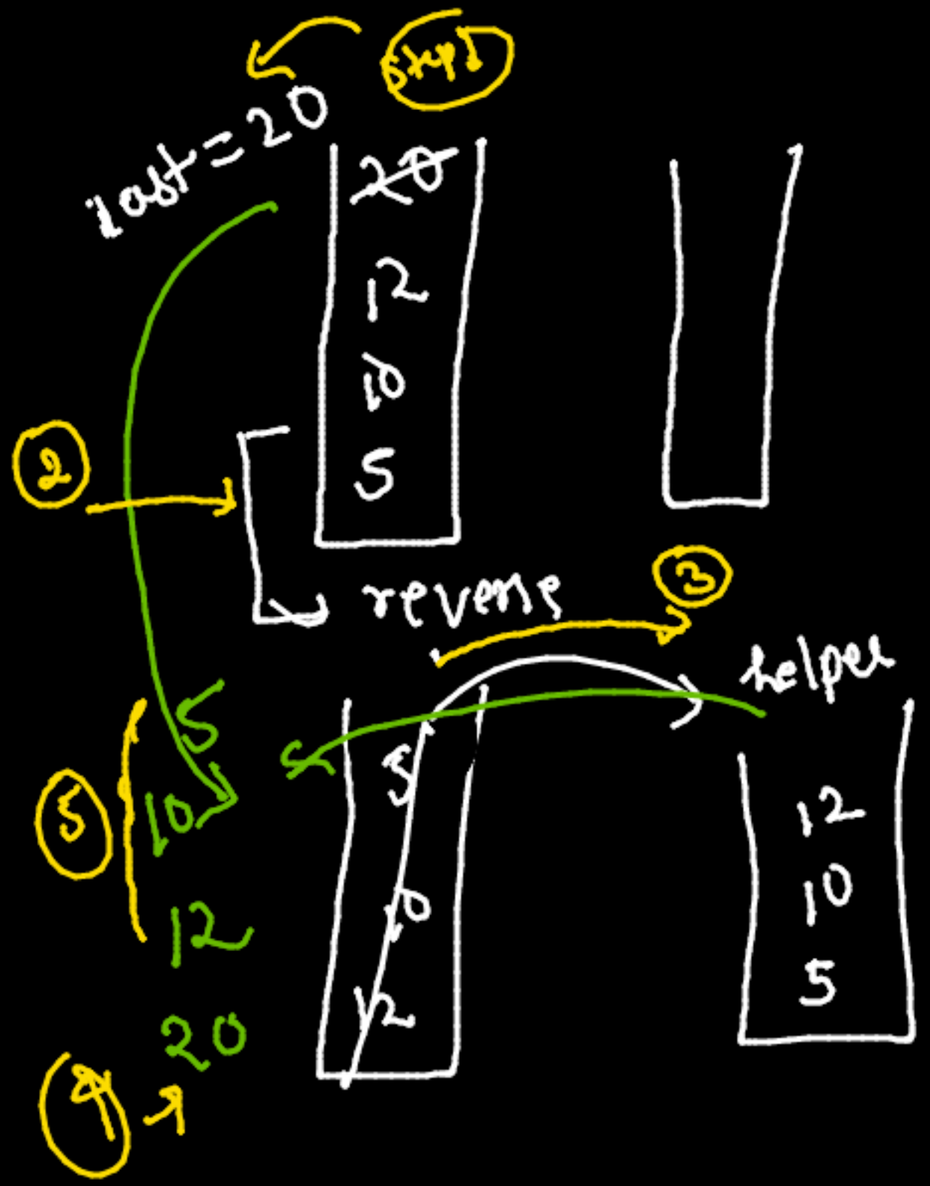
```
template <typename T>
class pair {
    T x;
    T y;
};

pair <int> p1;
pair <string> s1;
pair <char> c1;
```

```
template <typename T, template U>
class pair {
    T x;
    U y;
};

pair <int, string> p1;
pair <int, char> p2;
pair <char, string> p3;
```


Reverse a stack
3
1
2
5
10
12
15
helper1
helper2
s




```
void reverse( input, extra ) {        // stack
    if(input.size <= 1) return;
①   int last = input.pop();
②   reverse(input, extra);
③   while (!input.empty()) {
        int n = input.pop();
        extra.push(n);
    }
④   input.push(last);
⑤   while(!empty.size) {
        itn = empty.pop()
        input.push(n);
    }
```

false → ( ))

Balanced Parenthesis

( ) ( ) → true ✓

((( ))) → true ✓

(((( )))) → false

```
if ( c.empty() )
    return true

return false;
```

```
bool balanced (string s) {

  stack <char> c;

for (int i=0; i< s.size(); i++){

  if ( c.empty() && s[i] == ')')
          return false;

  elseif ( s[i] == '(') {
          c.push ('c');
  }
  else if ( s[i] == ')') {
          if ( c.top() == '(')
                  c.pop();
  } else c.push('(')
```

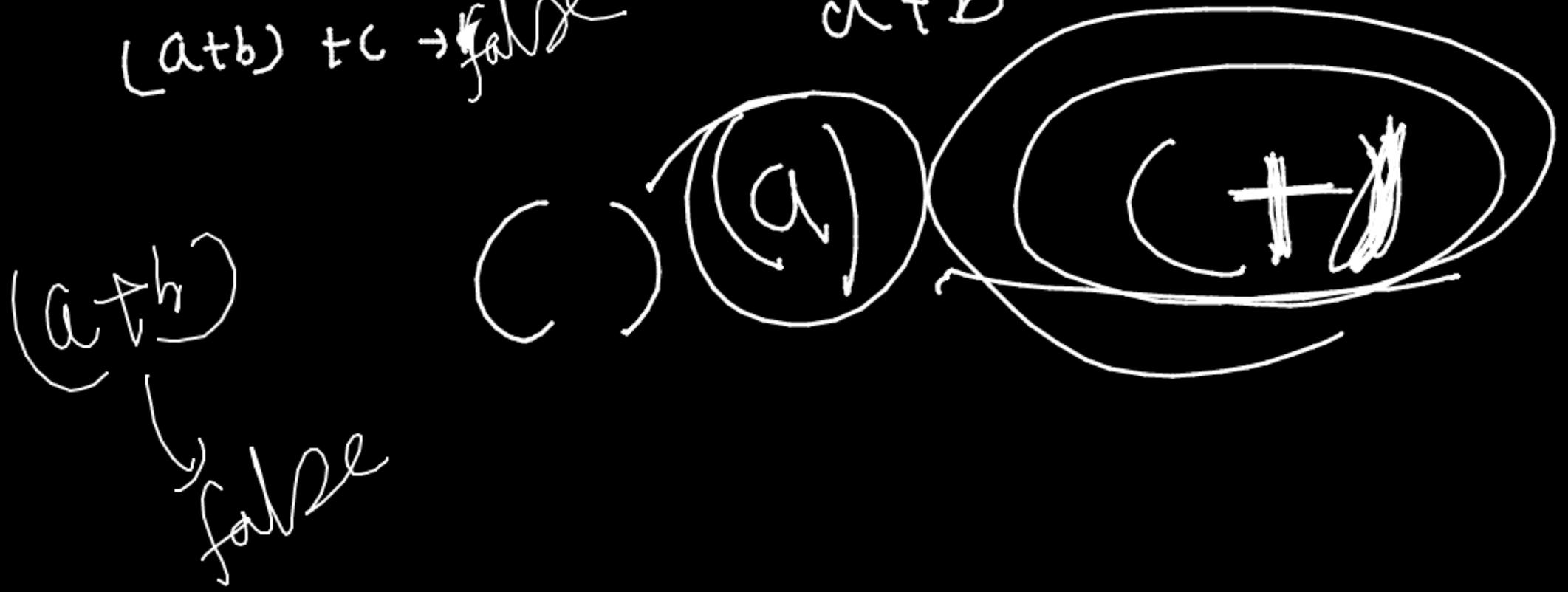
Redundant Bracket

eg $a+(b)+c \rightarrow$ true

$(a+b)+c \rightarrow$ false

$(a+b)$ $\quad$ $(c+$

$a+b$

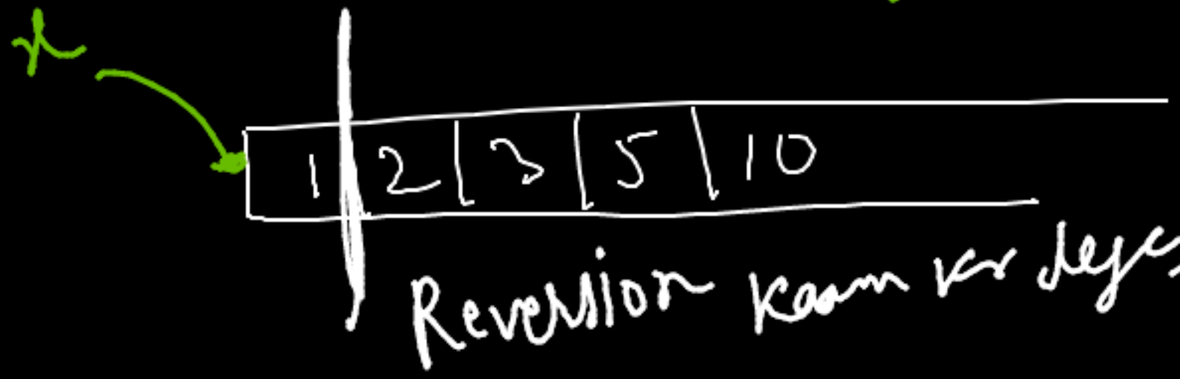


$(a+b)$ → false

Reverse Queue

FIFO

10 5 3 2 1



10 5 3 2 1

```
void reverse ( queue<int > &q ){
if (q.size()<=0)
    return

int x= q.top()
q.pop()

reverse (q);

q.push(x);
```

Maxm Bracket Reversal

① {{{{ → 2

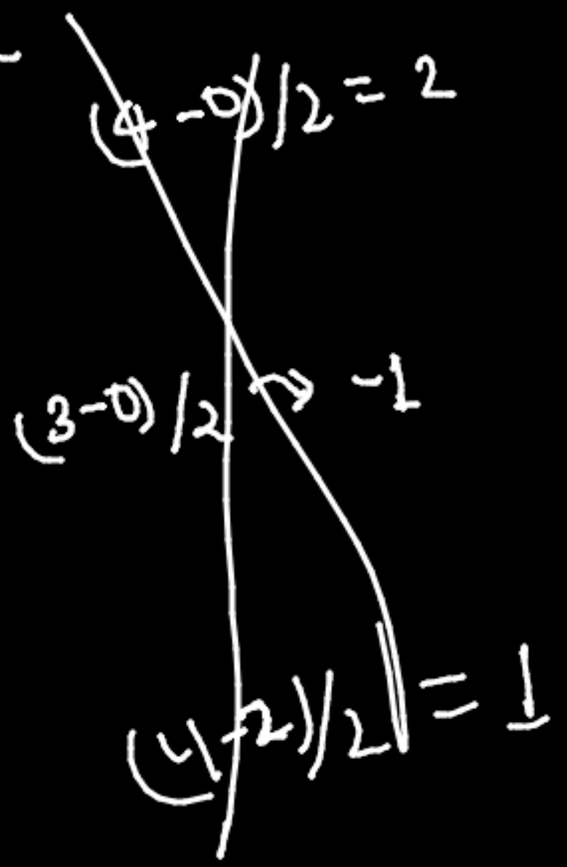
x = 4 y = 0

② {{{ → −1

x = 3 y = 0

{{{{}}}

x = 4 y = 2



}{ ⇒ 2

{}

Stock Span Problem