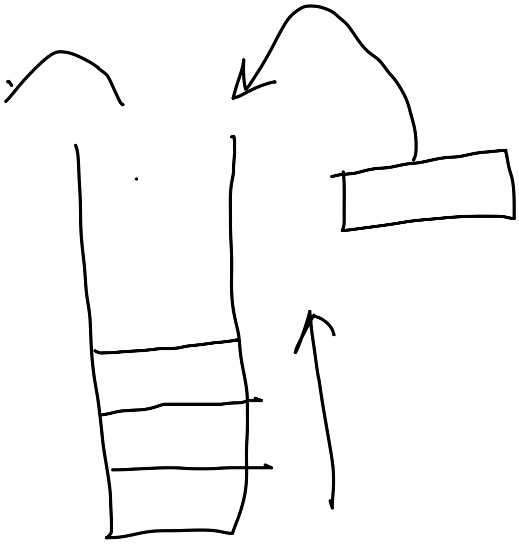
Stoick

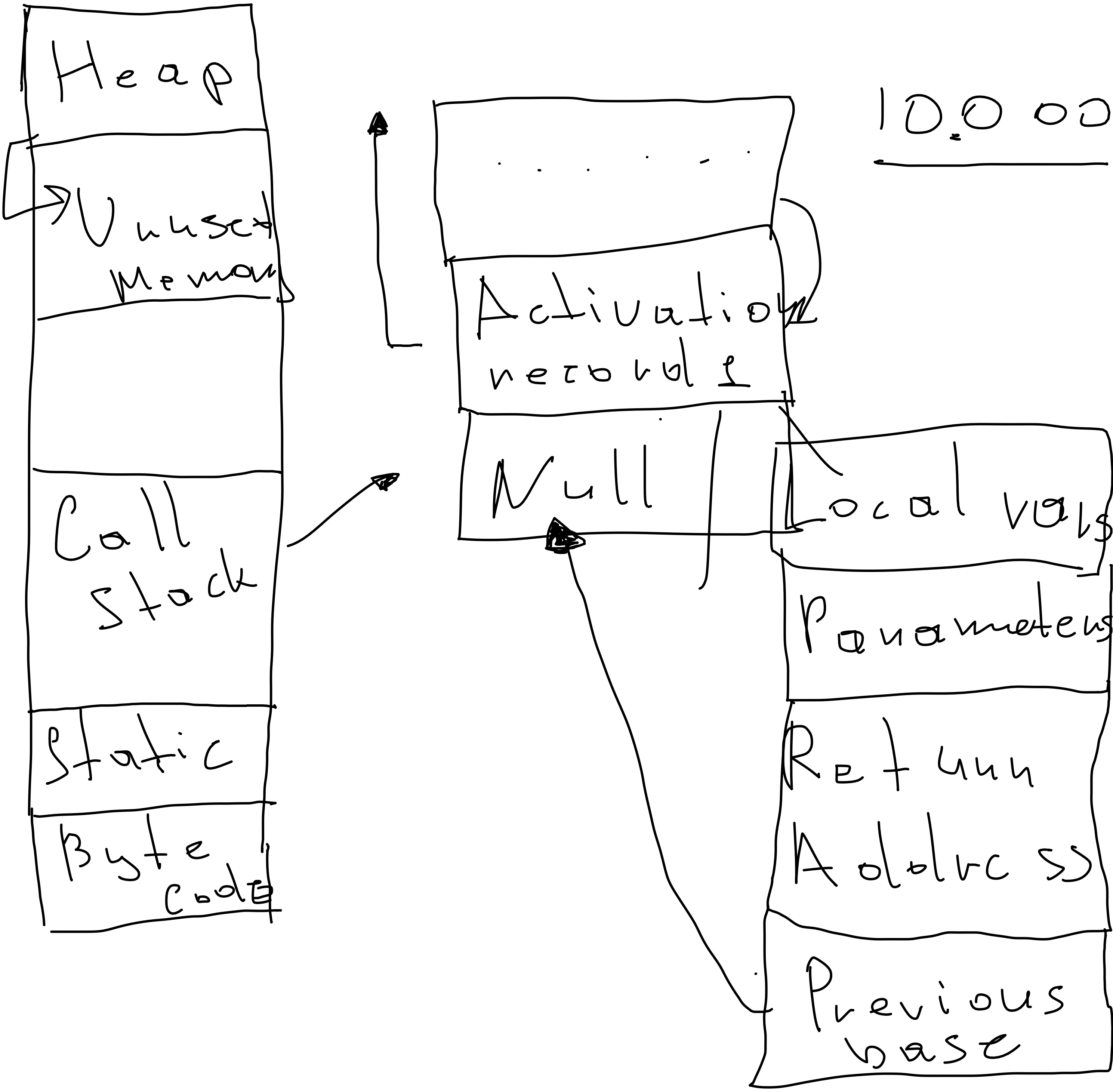
- Stack
- Applications
- Implementation
- Keeping minimum in Stack
- Brockets Seoquence

Stack

LIFO - Lost in First Out



- push(x)
- pop()


Heap
Unused memory
Call stack
Static
Byte code
. . . .
Activation record 1
Null
Local vars
Parameters
Return Address
Previous base
10.0 00

2 Application

↙

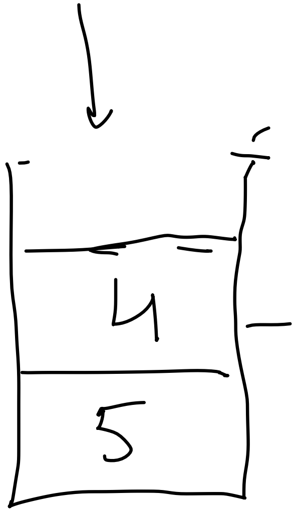
$(5+(((143 * 555)+3))\cdot(8+7)=$

→

Inverse Polish Notation

$53+ = 5+3$

$\underline{531\,\bigcirc\!\!\!\!+\;+}$



$1+3=4$

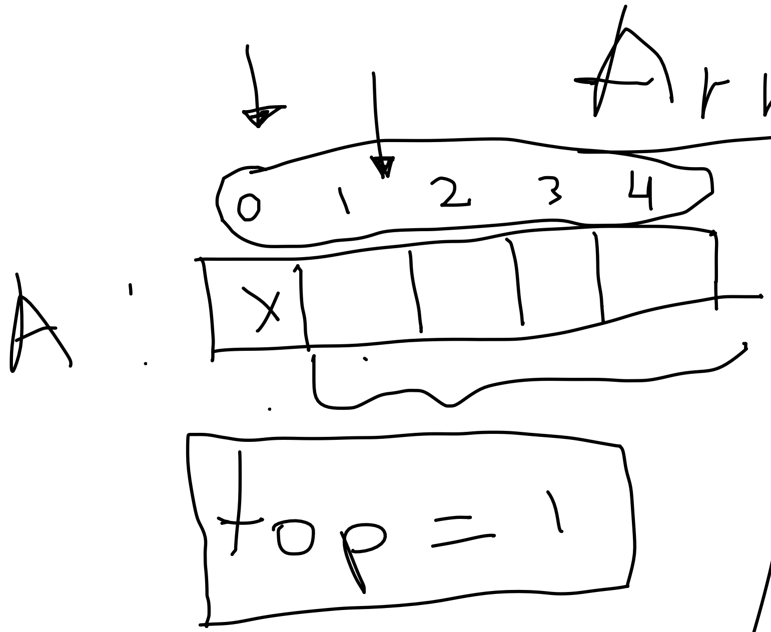
$4+5=9$

$\underline{531\,*+} = (3\cdot1)+5$

Implementation

- Array →
- Linked List →

Array

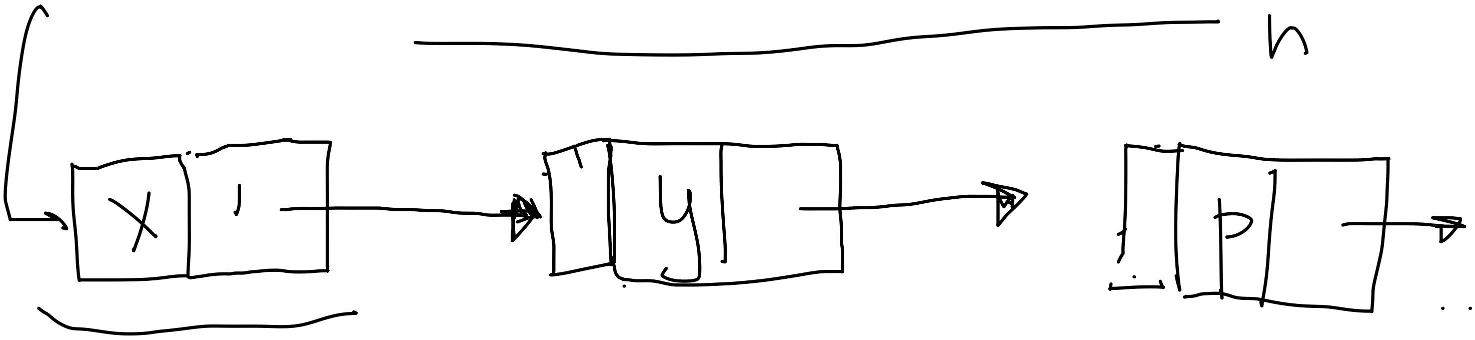


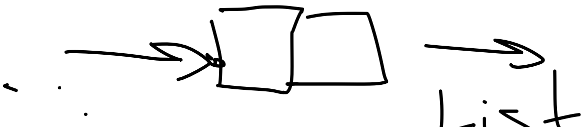
size = 5

```
Push(X)
if (top == size)
    → error
else:
    A[top] = X
    top += 1
```

```
Pop():
    if top==0:
        error
    else:
        top-=1
        return A[top]
```

Linked List





List

Read: O(n)

Append: O(1)

Array

O(1)

O(n)

Push

Head = null

capacity = x

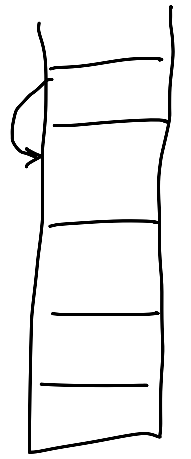
Push(x):

```
                h
                ↓
new_head = Node(x)
new_head.next = head
head = new_head
```

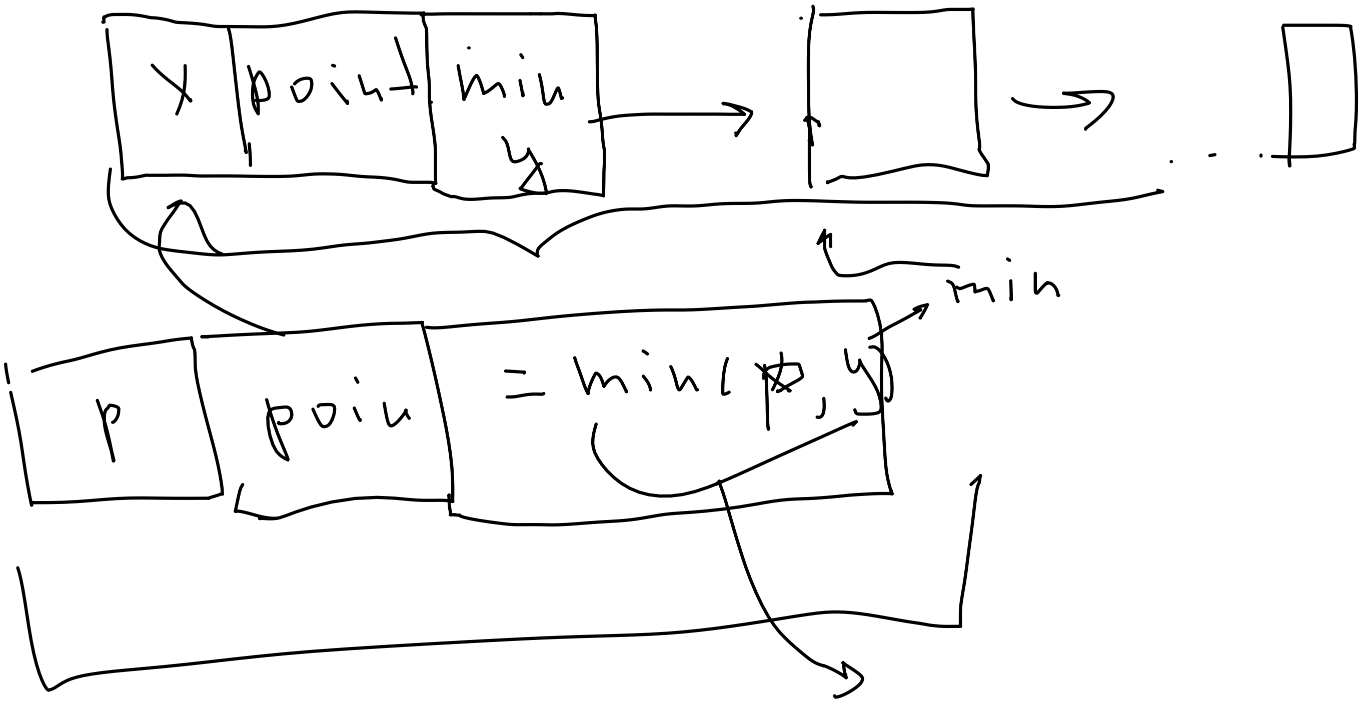
Pop():

```
old_head = head.
head = old_head.next
return old_head.value
```

Keeping the minimum



GetMin():



Correct Bracket sequence parsing

$$S \qquad \underbrace{[,\ (,\ \{,\ \},\ ),\ ]} = A$$

1. $S = \emptyset$

2. $S \to [\ S\ ]$

$(\ S\ )$

$\{\ S\ \}$

3. $\underline{S \to S_1 S_2}$

$\underline{[\ (\ )\ ]}$ $\quad$ $\underline{[\ ]\ (\ )}$

$\underline{[\ )\ ]\ (}$

Problem: A sequence S of elements from A.

Is S a connect or not?

([])

↓

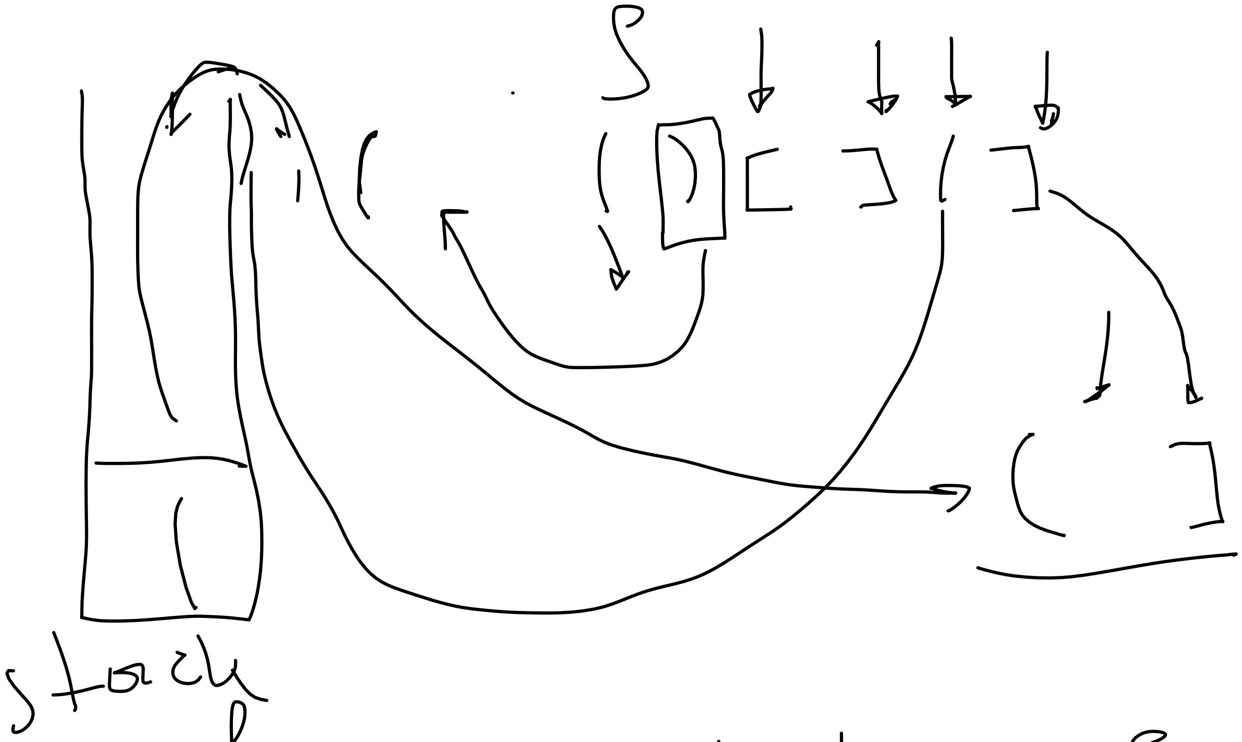
(][)

counter

(, [, { → +1

),], } → -1

counter < 0



```
for symbol is S:
    if symbol = '(', '[', '{'
        stack.push(symbol
    else:
        if (stack.empty):
            return NO
        if symbol = ')'
        && stack.top() = '[';
```

```
                retun NO

        if symbol = ']' {
            stack.top = '(' :
                return NO

        stack.pop()

O(S)
```