

Notations

Prefix

Infix

Postfix

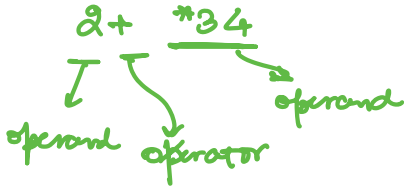
$$2 + 3 * 4$$

$$2 + 3 * 4$$

$$2 + 3 * 4$$

$$2 + 34^*$$

$$234^*+$$



operand operator operand

$$2 + 12$$

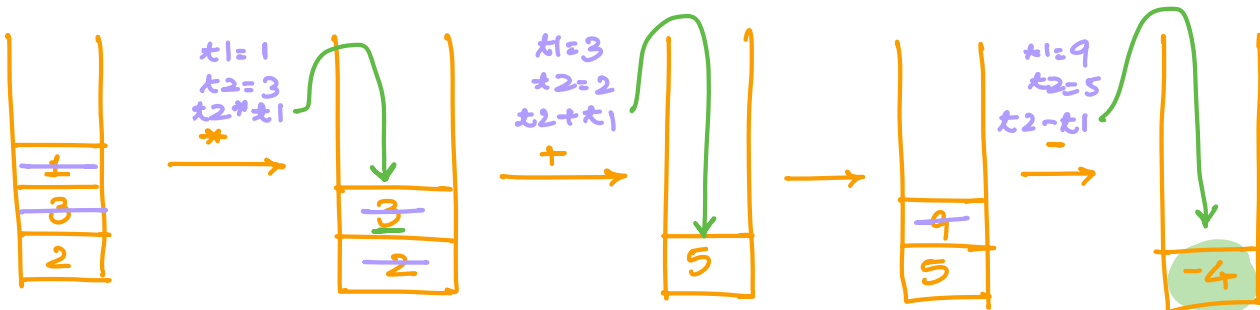
$$14$$

$$+ 2 * 34$$

Postfix Expression Evaluate

$$231^*+9-$$

↑↑↑ ↑ ↑↑



$$"012"$$

$$231^*+9-$$

'2' char → 2 int

$$'2' - '0'$$

↓

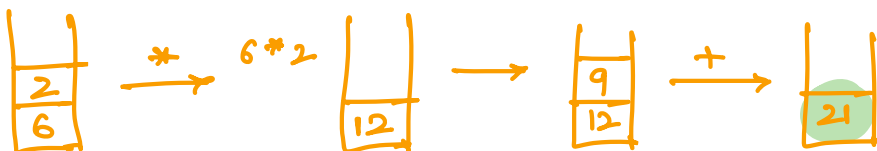
2 - 0

$$(2+2) - 2 = \underline{\underline{2}}$$

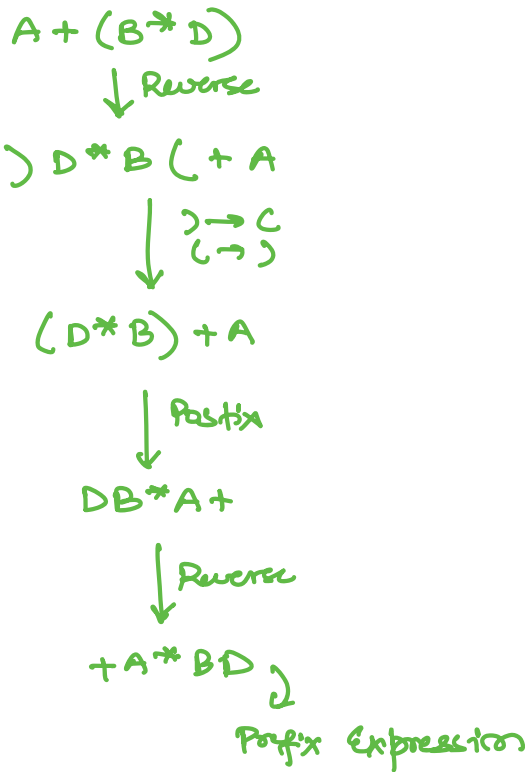
- '0' x
- '1' x+1
- '2' x+2
- '3' x+3
- '4'
- ⋮

Prefix Expression Evaluation

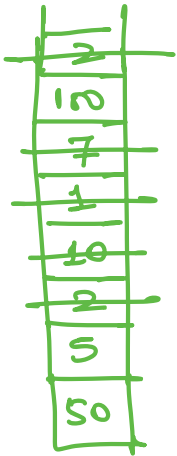
$$+9^*26$$



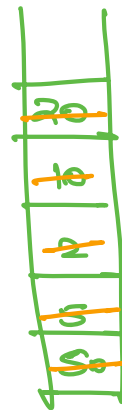
Infix \rightarrow Prefix Conversion



Stack Minimum



min? 1
 pop
 pop
 min? 2
 pop
 pop
 min? 5
 push(100)
 min? 5
 push(2)
 pop
 min? 5

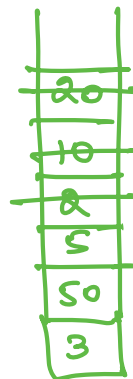


A1

min?
 push
 $20 \checkmark$ $10 \checkmark$ $2 \checkmark$ $5 \checkmark$ $50 \checkmark$
min? (2)

$O(n) : TC$
 $O(n) : SC$

A2



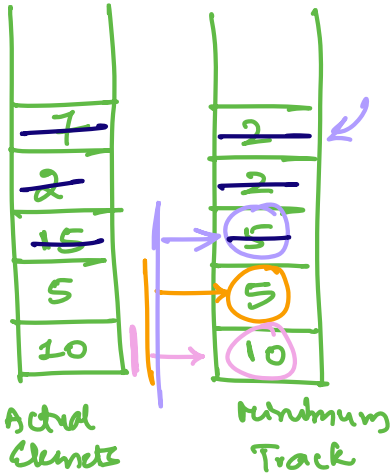
min = 2
 pop()
 pop()
 min? 2
 pop()
 min?
 keep a track while pushing the element

A3

→ new stack for maintaining mins

TC: O(1)

SC: O(n)



Push(10)	min?
Push(5)	2
Push(15)	pop
Push(2)	pop
Push(7)	min? 5
	pop

`min_track.push(min(item, min_track.top()))`

7 2

2

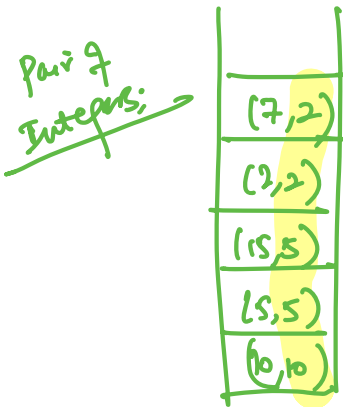
A4

TC: O(1)

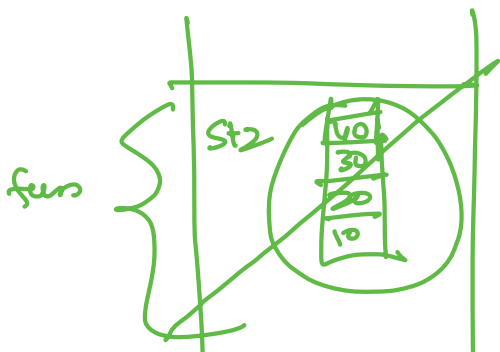
SC: O(1)

homework

Intuition: Change the values of stack



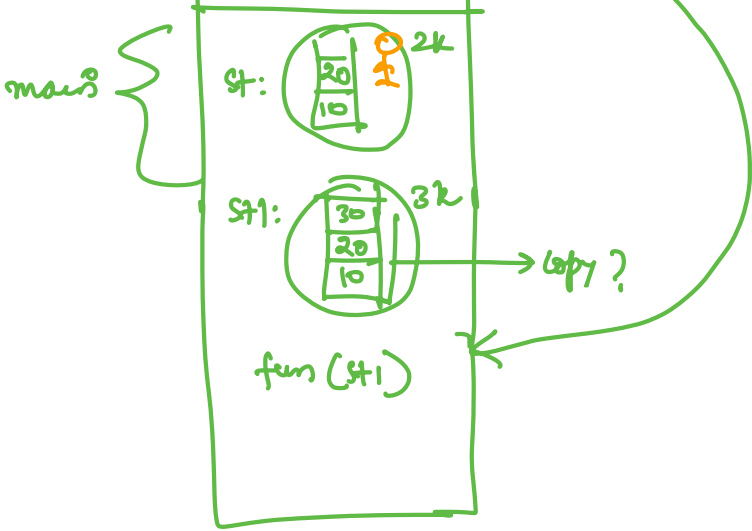
Stack Pass:



```

void fun(int a)
{
    cout << a;
}
int main()
{

```



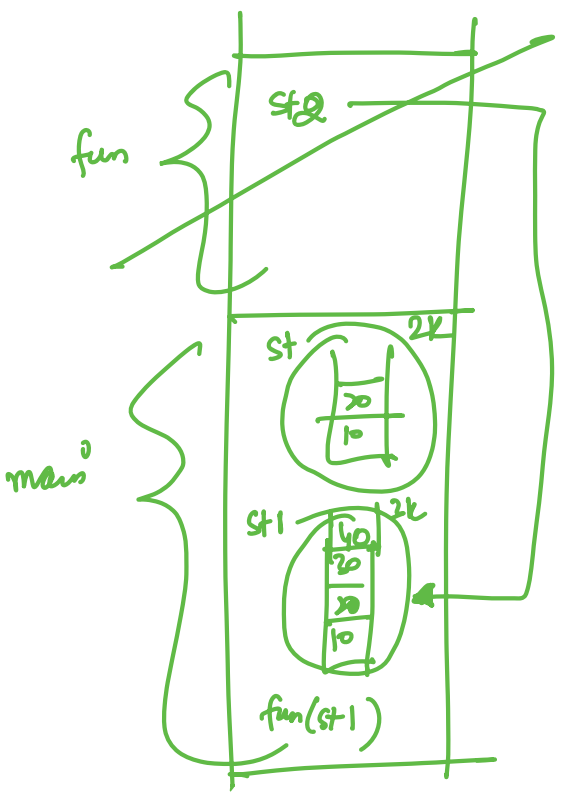
```

    {
    }
    LHS    RHS
    int a = 2
  
```

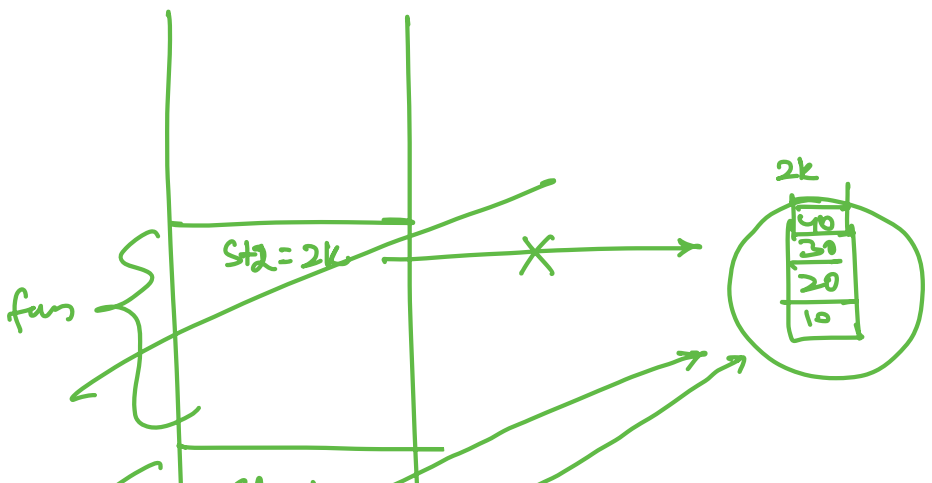
Call: fun(2);

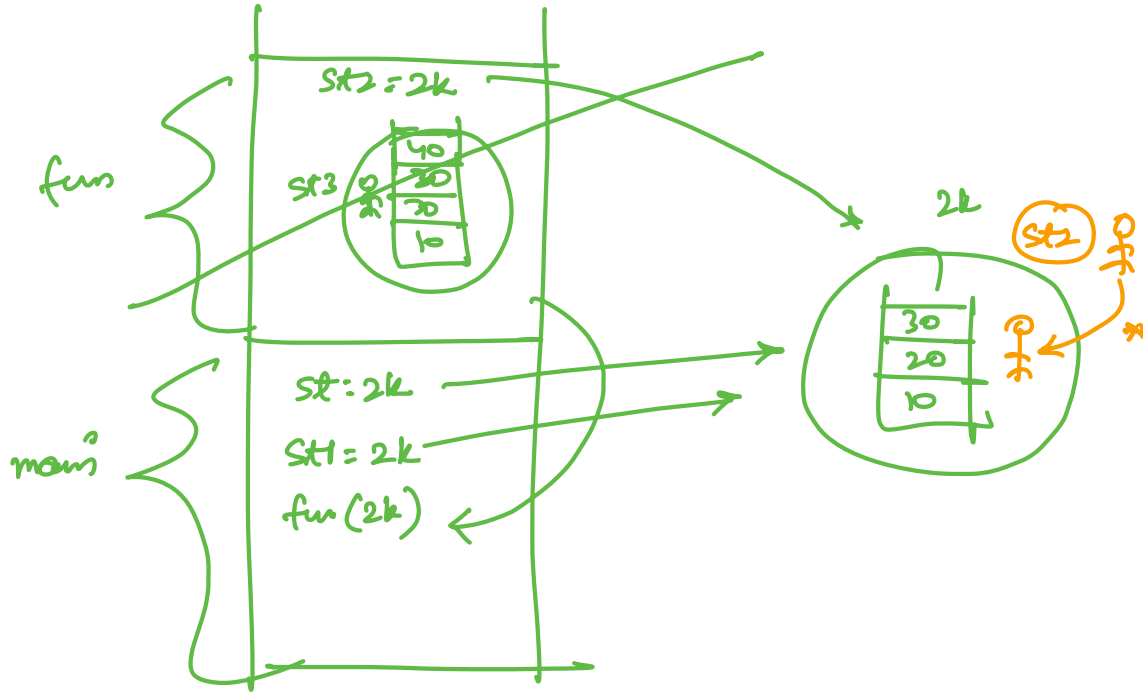
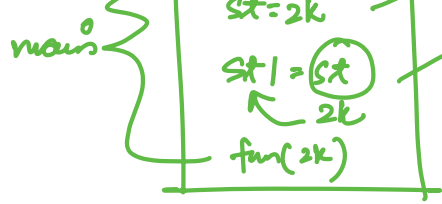
→ RHS

stack(int * st2 = st1);



new





Stack Reverse Display



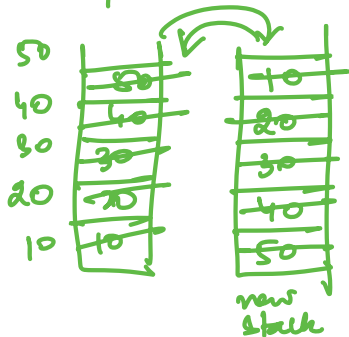
Normal: 50 40 30 20 10

Reverse: 10 20 30 40 50

- Output

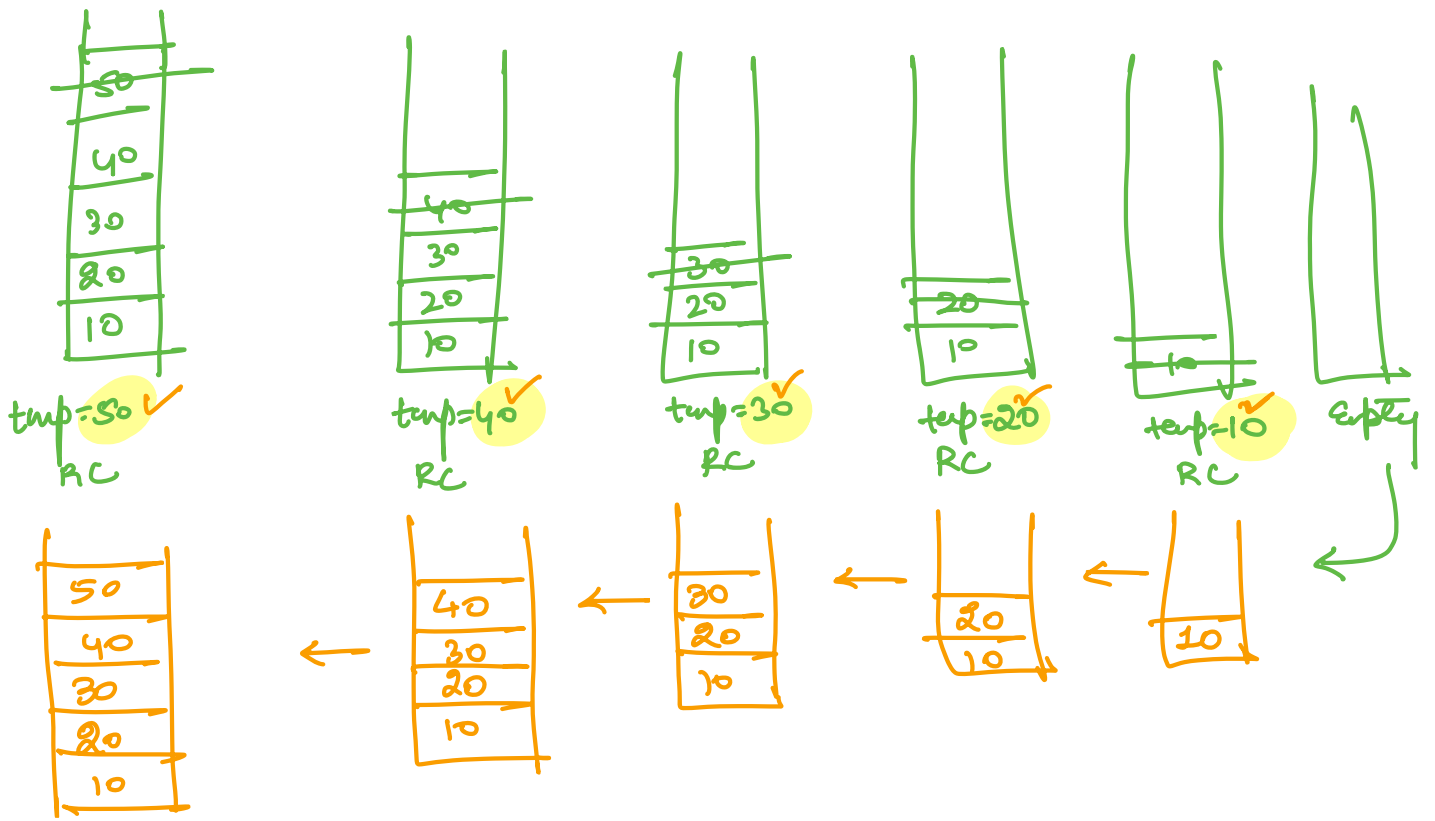
- Stack should remain same.

Extra space Allowed:

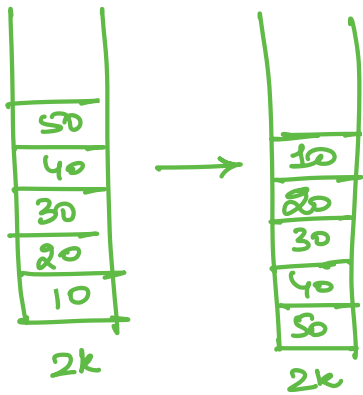


10 20 30 40 50

(Explicit Extra Space)

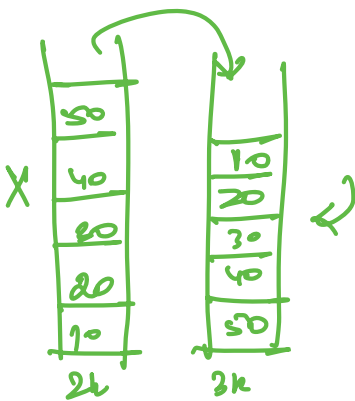


Stack Actual Reverse



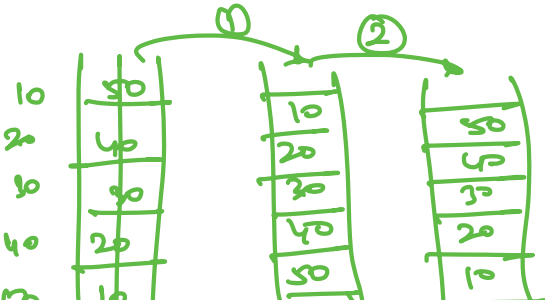
In some stack values should come in reverse order

(A1)

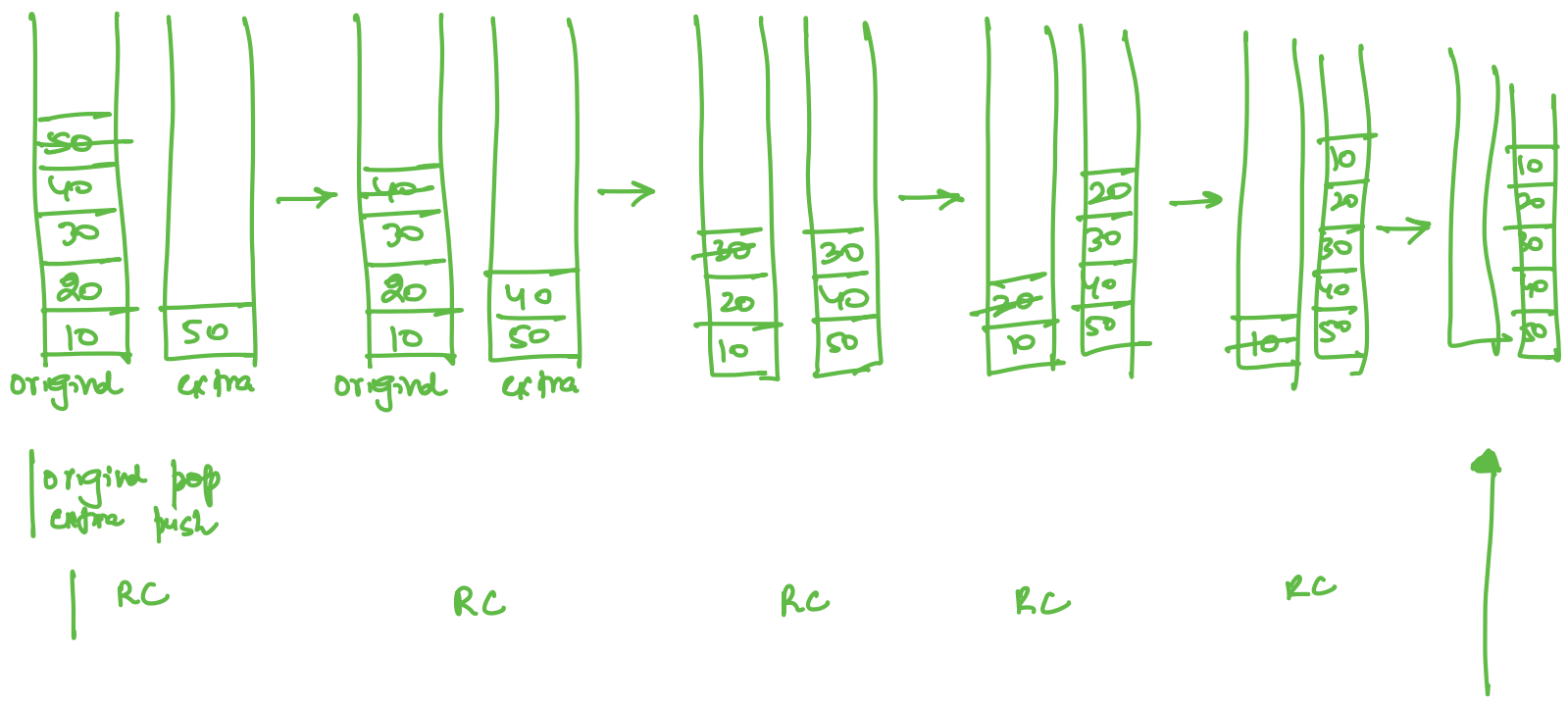


Original stack changes X

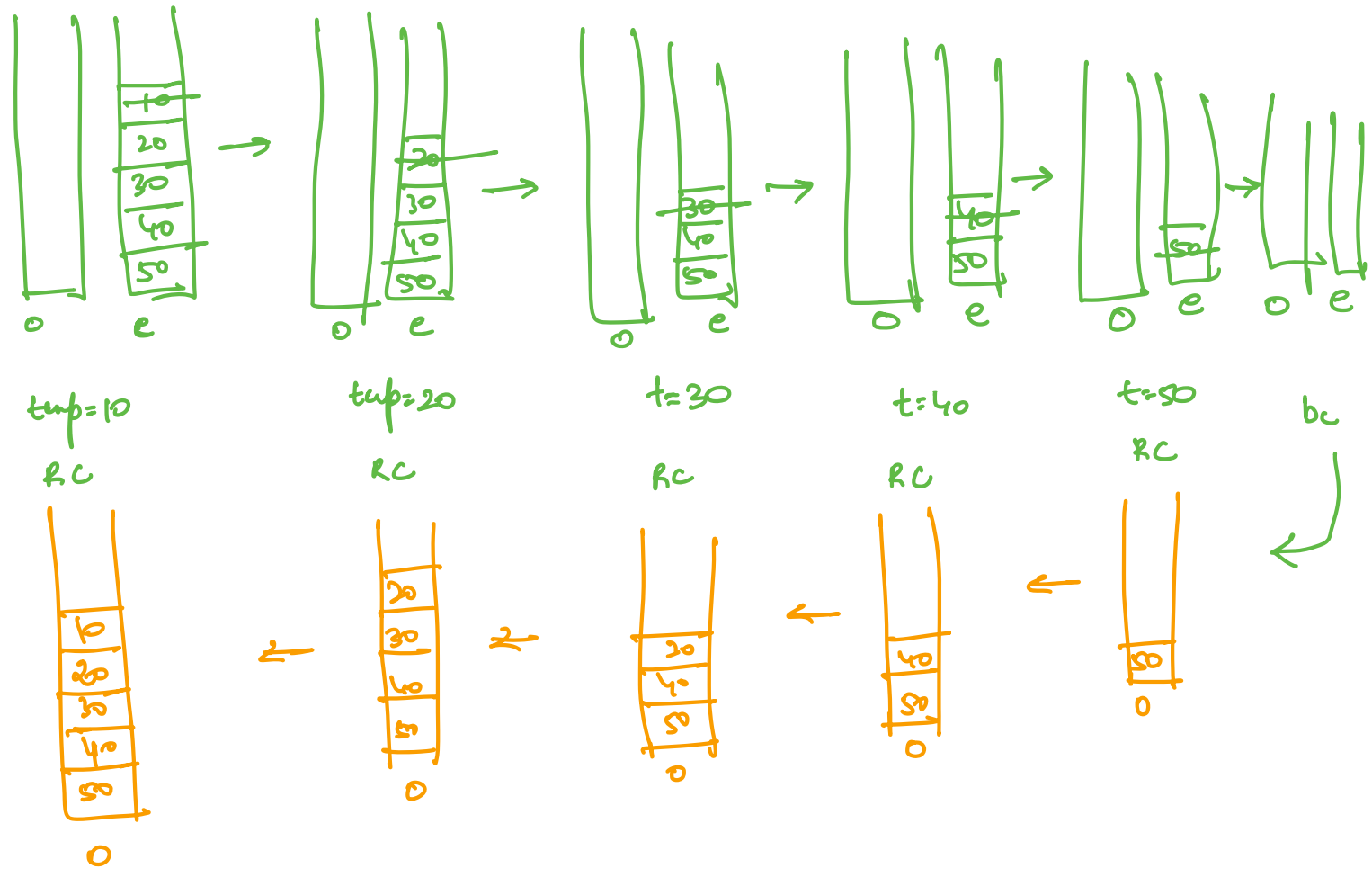
(A2)



2 Extra Stack

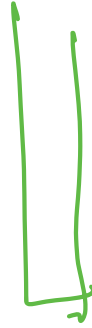
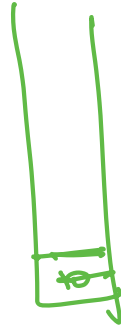
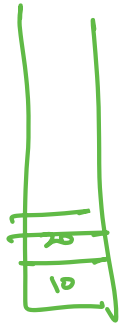
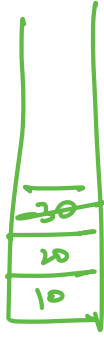


Original pop
Extra push



Display:

Order: 40 30 20 10



40

30

20

10

empty.

