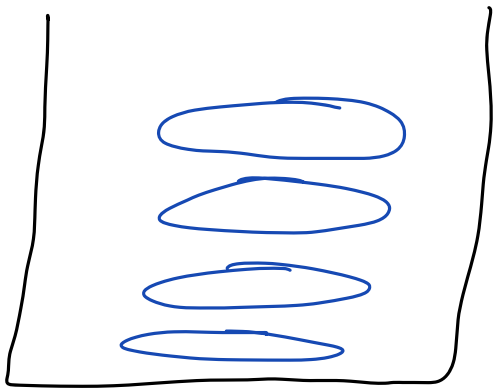
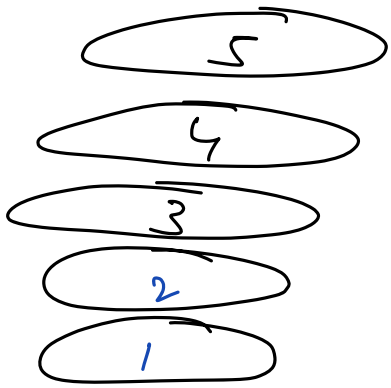
Stack Idli Cooker

DS





both insert & remove → top

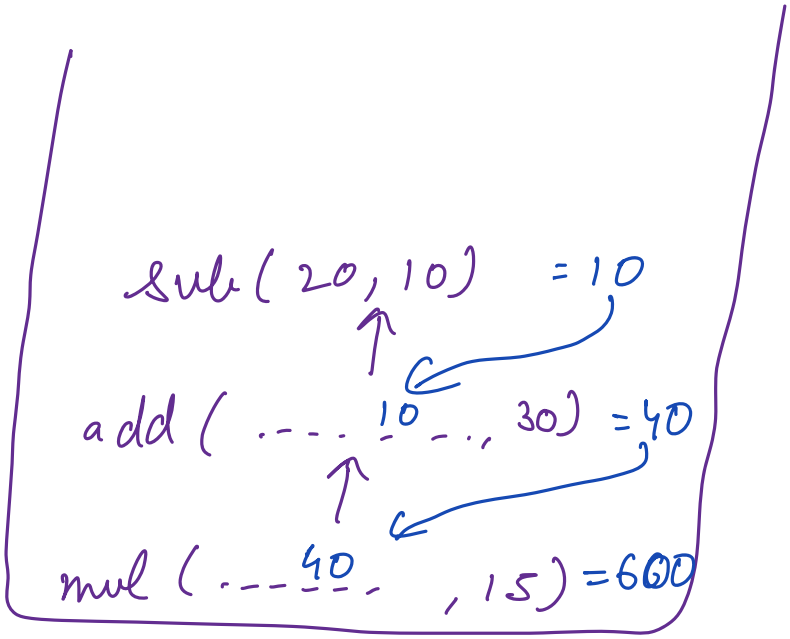
Call Stack →

```
int add(x,y){
|   return x+y
}
```

```
int sub(x,y){
|   return x-y
}
```

```
int mul(x,y){
|   return x*y
}
```

print (mul(add(sub(20,10),30),15))



Local machine ⇒ RAM

Online IDE's

Max ⇒ 10^7 integers

Types of memory in Java

1) Stack → all primitive data types
(Temp changes)

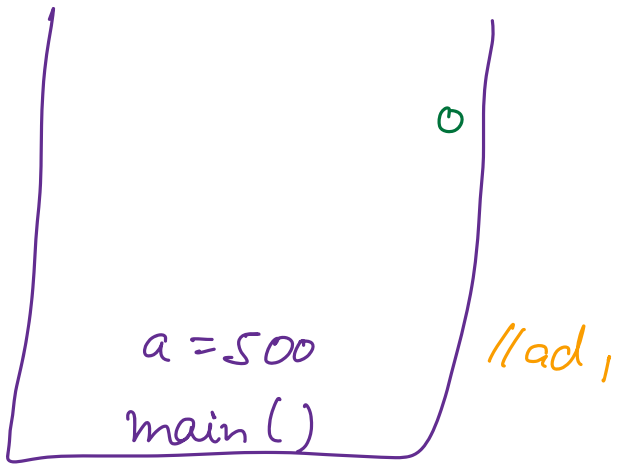
2) Heap → Arrays, Objects etc
(Changes are permanent)

⇒ anything declared using new keyword is on heap

integers are passed by value

arrays are passed by reference

```
int a = 500

change(a)

change(int a){
    a = 100
}

bprint(a)
```




```
ar[1] = {500}

change(ar)
print ar(0)

change(int[] a){
    a[0] = 100
}
```

Quiz

```
static void change (int a) {
    a = 50
}

public static void main (String args[]) {
    int a = 10
    change (a)
    print (a)
}
```

→ integer passed
→ stack
→ temp

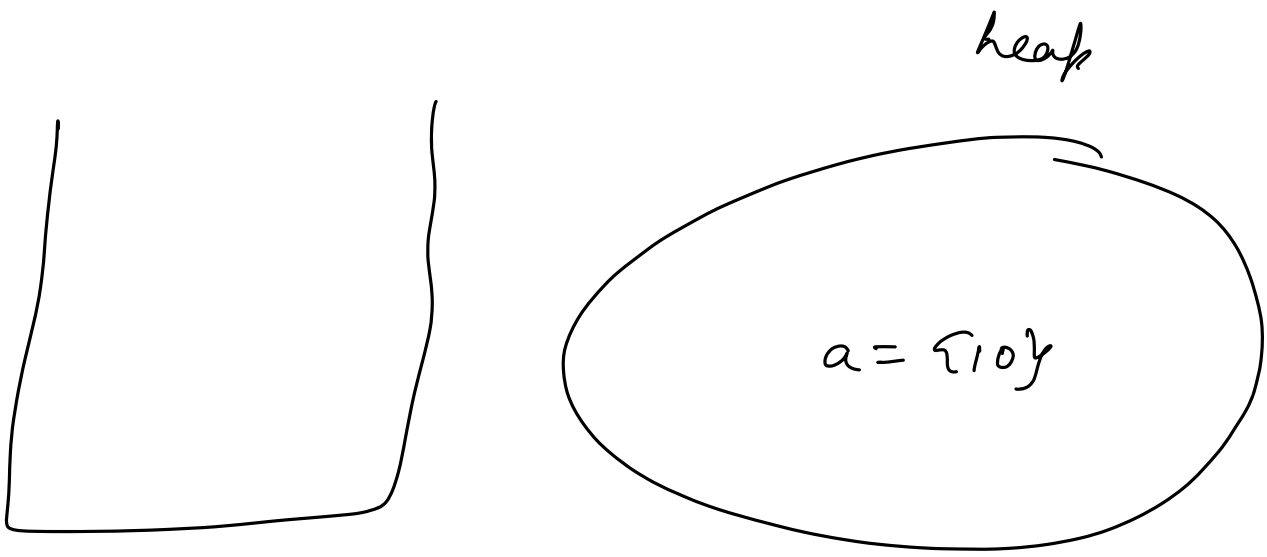
Quiz 2

```
static void change (int a[] ) {
    a[0] = 50
}

public static void main (String args[]) {
    int[] a = {10}
    change (a)
    print (a[0])
}
```

→ array passed
→ heap
→ permanent changes

```
static void change (int a[] ) {
    a = new int [1]
    a[0] = 50

}
```

```
public static void main ( String args[]) {
    int[] a = {10}
    change (a)
    print (a[0])
}
```



Learning

→ Anything declared using new, is only destroyed using
   a) Goes out of scope
   b) delete key word

```
static void change (int a[] ) {
    a = new int[1]
    a[0] = 100
}

public static void main (String args[]) {
    int[] a = {10, 20, 30}
    change (a)
    print (a[0])
}
```

same as above
will not change

```
static void swap (int a, int b){
        temp = a
        a = b
        b = temp
}

public static void main (String args[]){
        a = 10          b = 20
        swap (a, b)
        print (a)       print (b)
}
```

→ integers
→ stack
→ temp changes

```
static void swap (int[]a ,int[]b){
        temp = a[0]
        a(0) = b[0]
        b[0] = temp
}

public static void main (string args[]){
        a ={10}        b ={20}
        swap ( a, b)
        print (a)          print (b)
}
```

arrays
permanent changes

```
static int[] fun ( int[] a ) {
    a = new int[2]
    a[0] = 50        a[1] = 60
    return a
}

public static void main (String args[]) {
    a = {10, 20, 30}
    a = fun(a)
    print a[0]
}
```

a = {10, 20, 30}

a = {50, 60}

a = fun(a)

a = fun({10, 20, 30})

a = {50, 60}

```
static void test ( int[] a ) {
    a = new int[2]
    a[0] = 94
}

public static void main ( string args[] ) {
    a = {10, 20, 30}
    test (a)
    print a[0]
}
```

similar to prev quiz

array unchange

{done}

Sorting → application ✓

→ algorithms ✗

$i = 3$ $\quad\quad$ $m = 8$

$x = 3$ $\quad\quad$ $y = 7$

3, 7
4, 6
5, 5
6, 4
7, 3
8, 2
9, 1
10, 0
11, -1

3×3

$a[3][3] =$ { 1 2 3 / 4 5 6 / 7 8 9 }

$a[9] =$ { 1, 2, 3, 4, 5, 6, 7, 8, 9 }