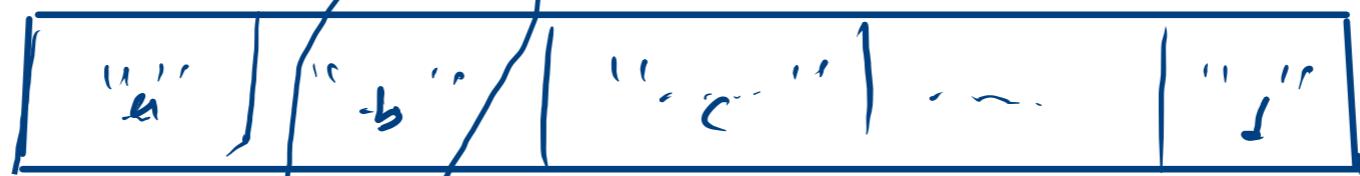


Class  
Object

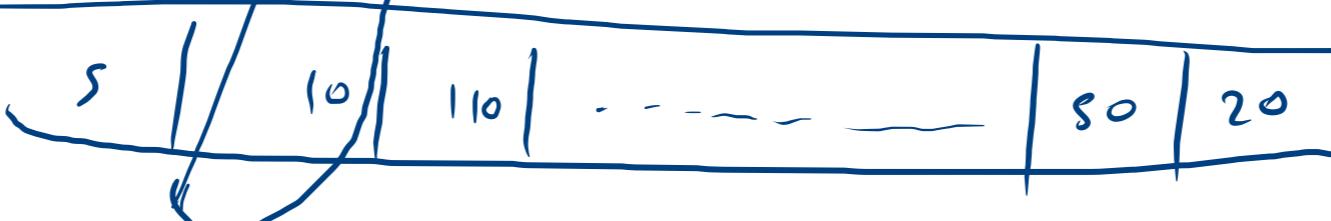
OOP

names =



i

age =



name[i], age[i]

Person  
int  
name  
age



class

```
class Person {  
    String name  
    int age
```

}

```
Person p1 = new Person();
```

```
p1.name = "b";
```

```
p1.age = 20;
```

b=a

a s  
b ≠ s

Stack

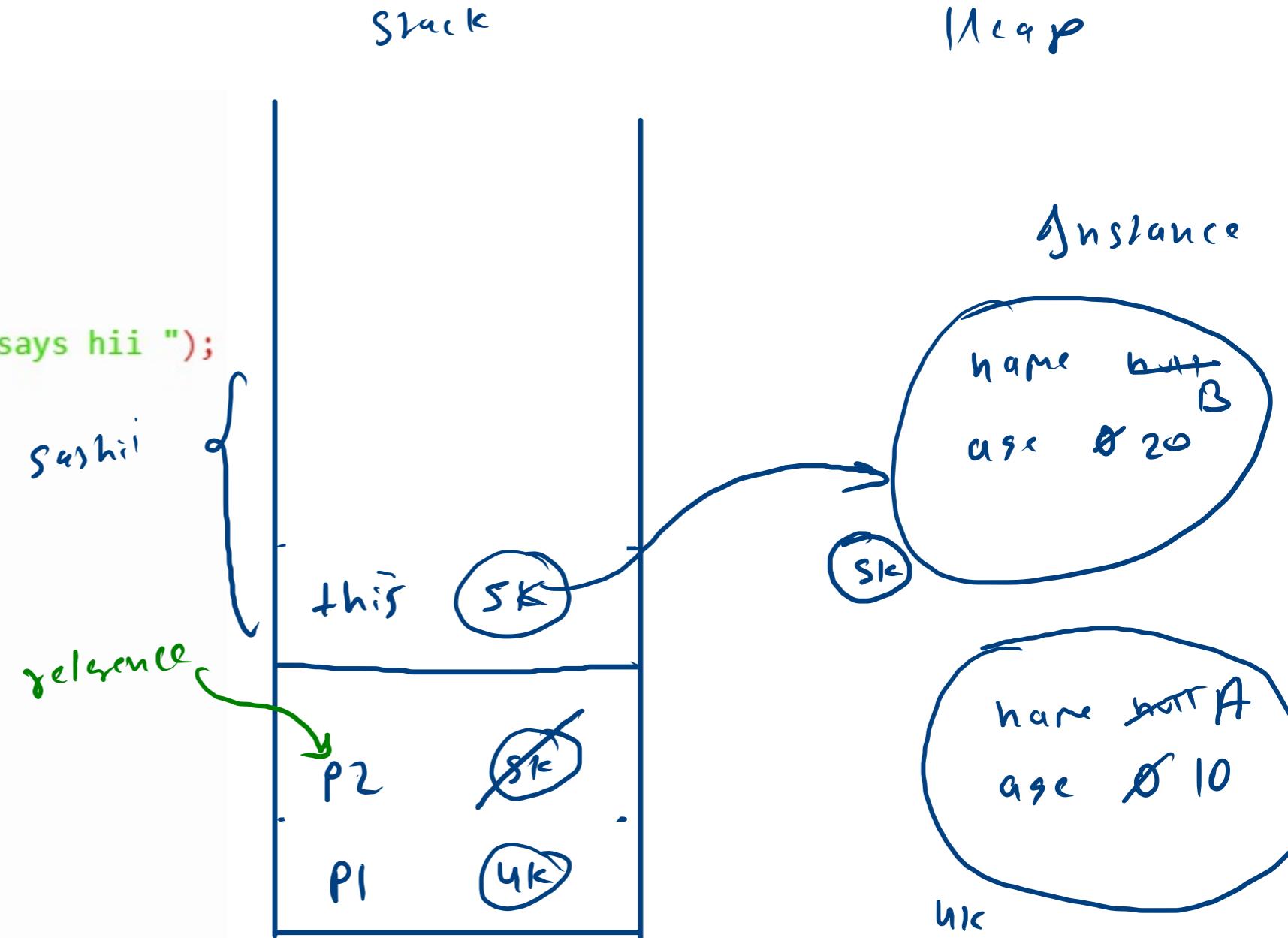
Heap

```
public static class Person{  
    String name;  
    int age;  
  
    public void sayHii(){ B  
        System.out.println(name+" "+ age+" says hii ");  
    }  
  
    public static void main(String[] args) {  
        Person p1 = new Person();  
        p1.name = "A";  
        p1.age = 10;  
  
        p1.sayHii();  
  
        Person p2 = new Person();  
        p2.name = "B";  
        p2.age = 20;  
        p2.sayHii();  
    }  
}
```

p2 = p1

A 10 says hii

B 20 says hii



```

public static void main(String[] args) {
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 10;

    // p1.sayHii();

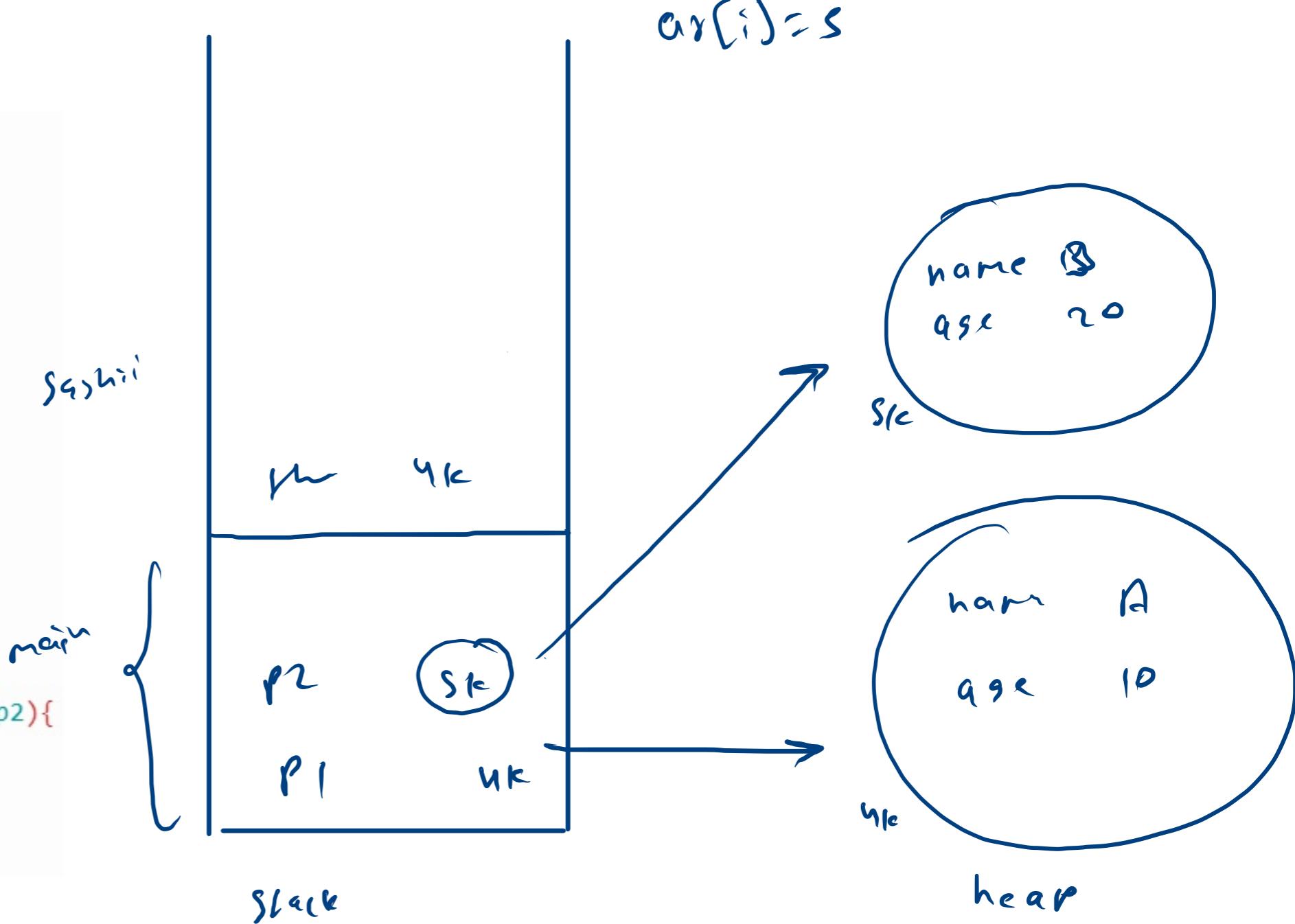
    Person p2 = new Person();
    p2.name = "B";
    p2.age = 20;
    // p2.sayHii();

    swap(p1, p2);
    p1.sayHii();
    p2.sayHii();
}

public static void swap(Person p1, Person p2){
    Person temp = p1;
    p1 = p2;
    p2 = temp;
}

```

A 10 sayHii  
B 20 sayHii

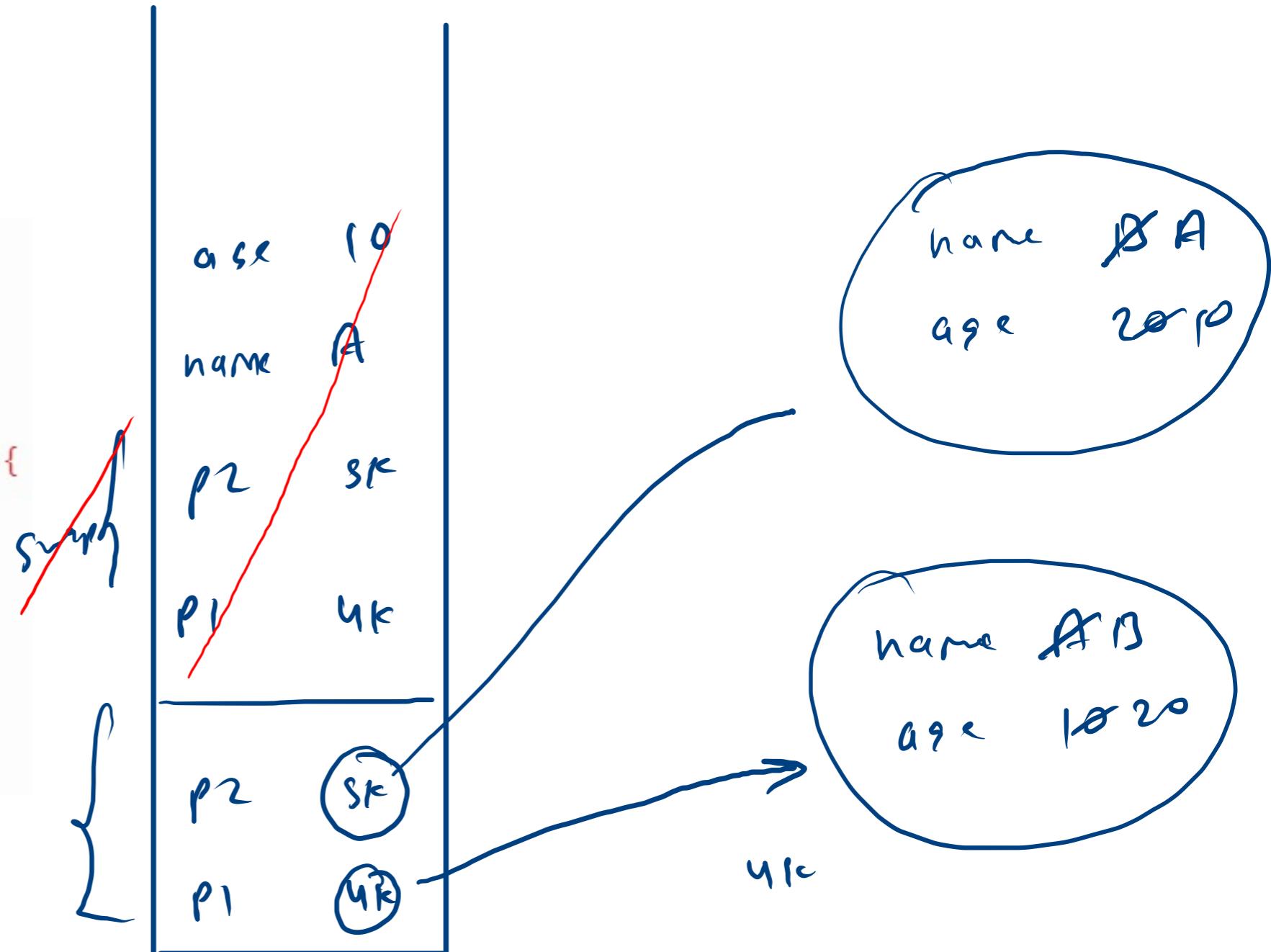


```
    swap(p1,p2);
    p1.sayHii();
    p2.sayHii();
}
```

```
public static void swap2(Person p1, Person p2){
    String name = p1.name;
    p1.name = p2.name ;
    p2.name = name;

    int age = p1.age;
    p1.age = p2.age;
    p2.age = age;
}
```

B 20 sashil  
A 10 sashil



```

    • swap3(p1,p2);
    • p1.sayHii();
    • p2.sayHii();
}

public static void swap3(Person ps1, Person ps2){
    • ps1 = new Person();

    • String name = ps1.name;
    • ps1.name = ps2.name ;
    • ps2.name = name;

    • ps2 = new Person();
    • int age = ps1.age;
    • ps1.age = ps2.age;
    • ps2.age = age;
}

```

JVM

4K

5K

use  
name

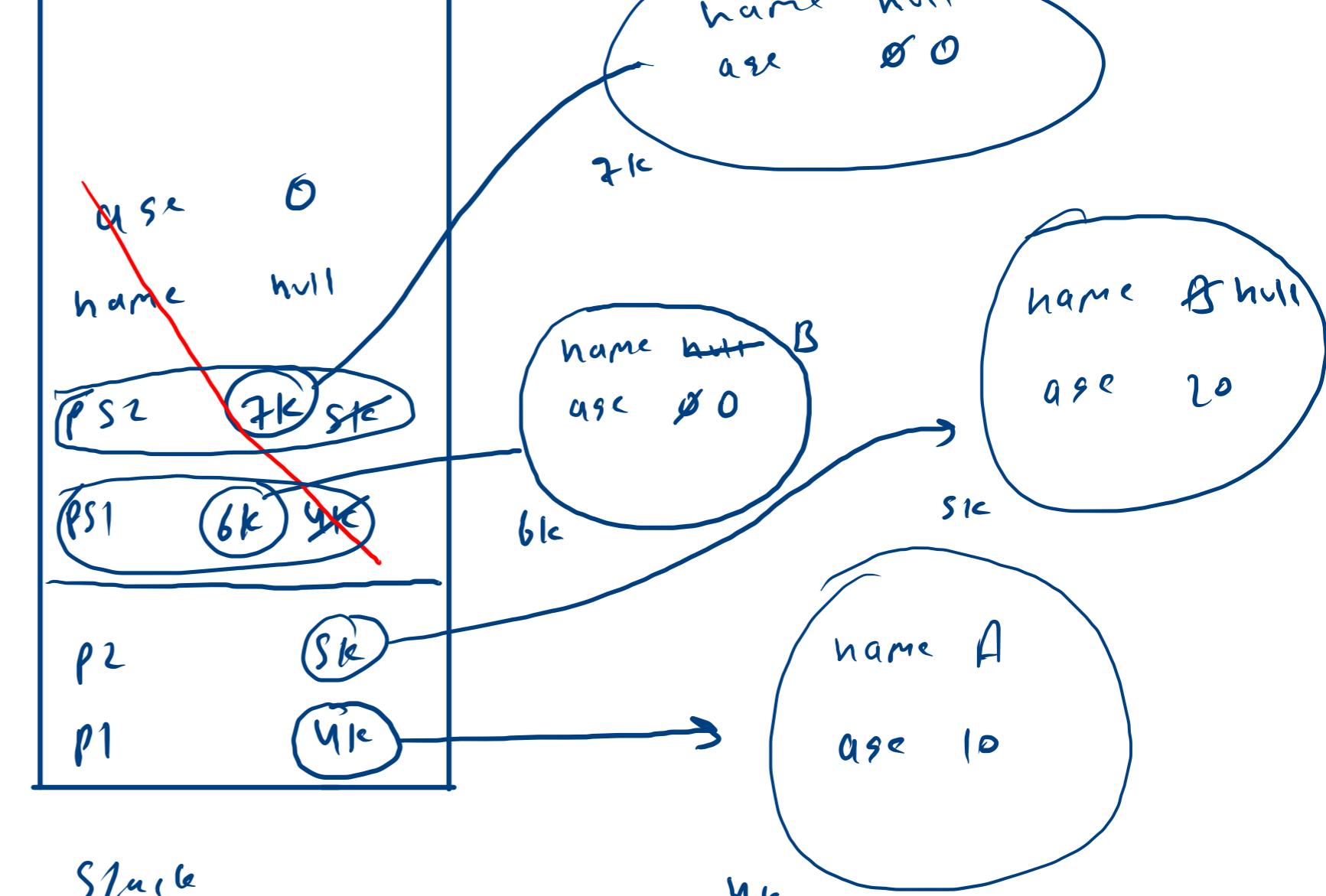
0

null



p2  
p1

Stack



```

Person(){
    // System.out.println("in const...");
}

// parameter... constructor
Person(String name, int age){
    this.name = name;
    this.age = age;
}

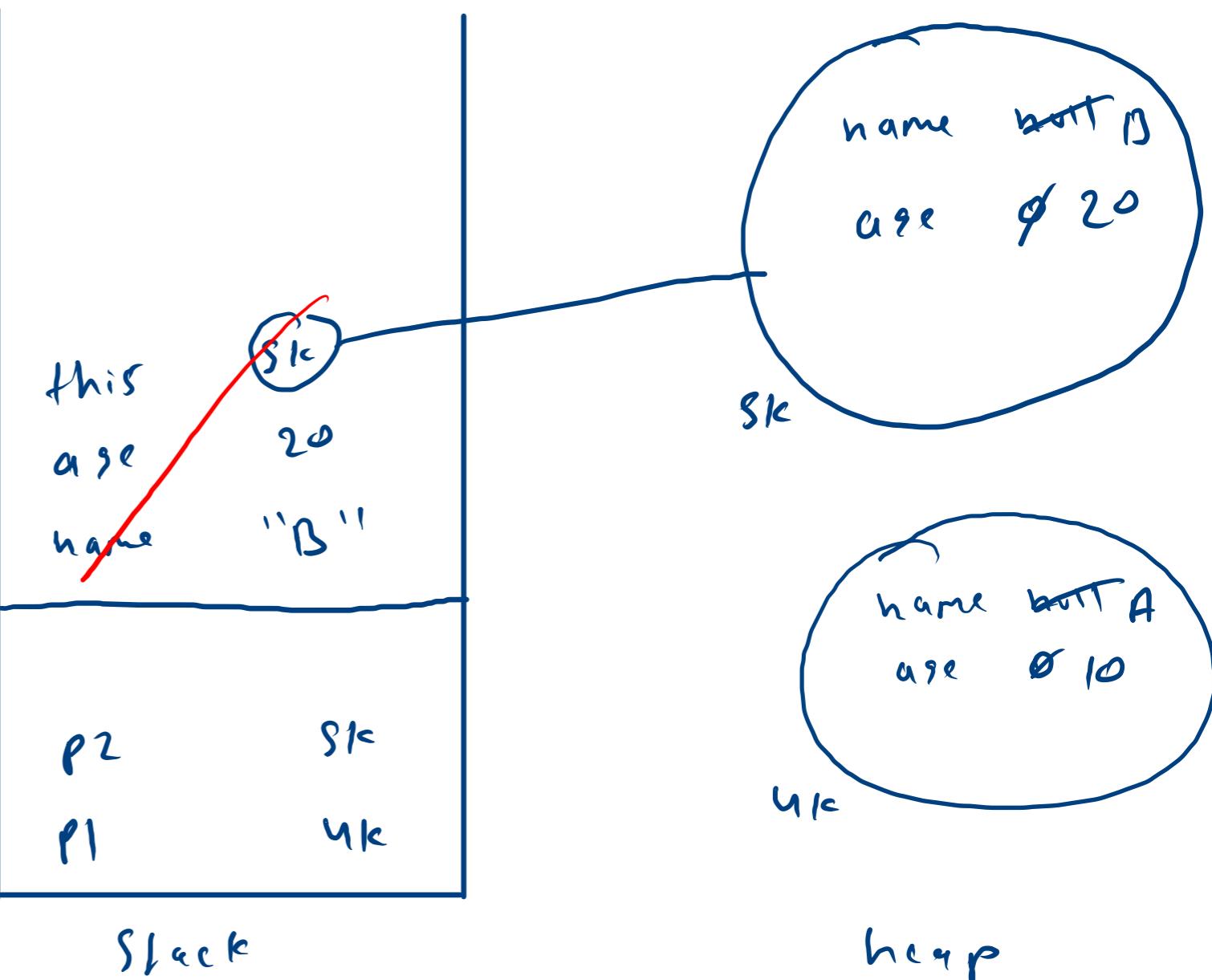
public void sayHii(){
    System.out.println(name + " " + age + " says hii ");
}

public static void main(String[] args) {
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 10;           A 10

    p1.sayHii();

    Person p2 = new Person("B", 20);  B 20
    // p2.name = "B";
    // p2.age = 20;
    p2.sayHii();
}

```



```

1 "B" 20
// parameter... constructor
Person(String n, int a){
    name = n;
    age = a;
}

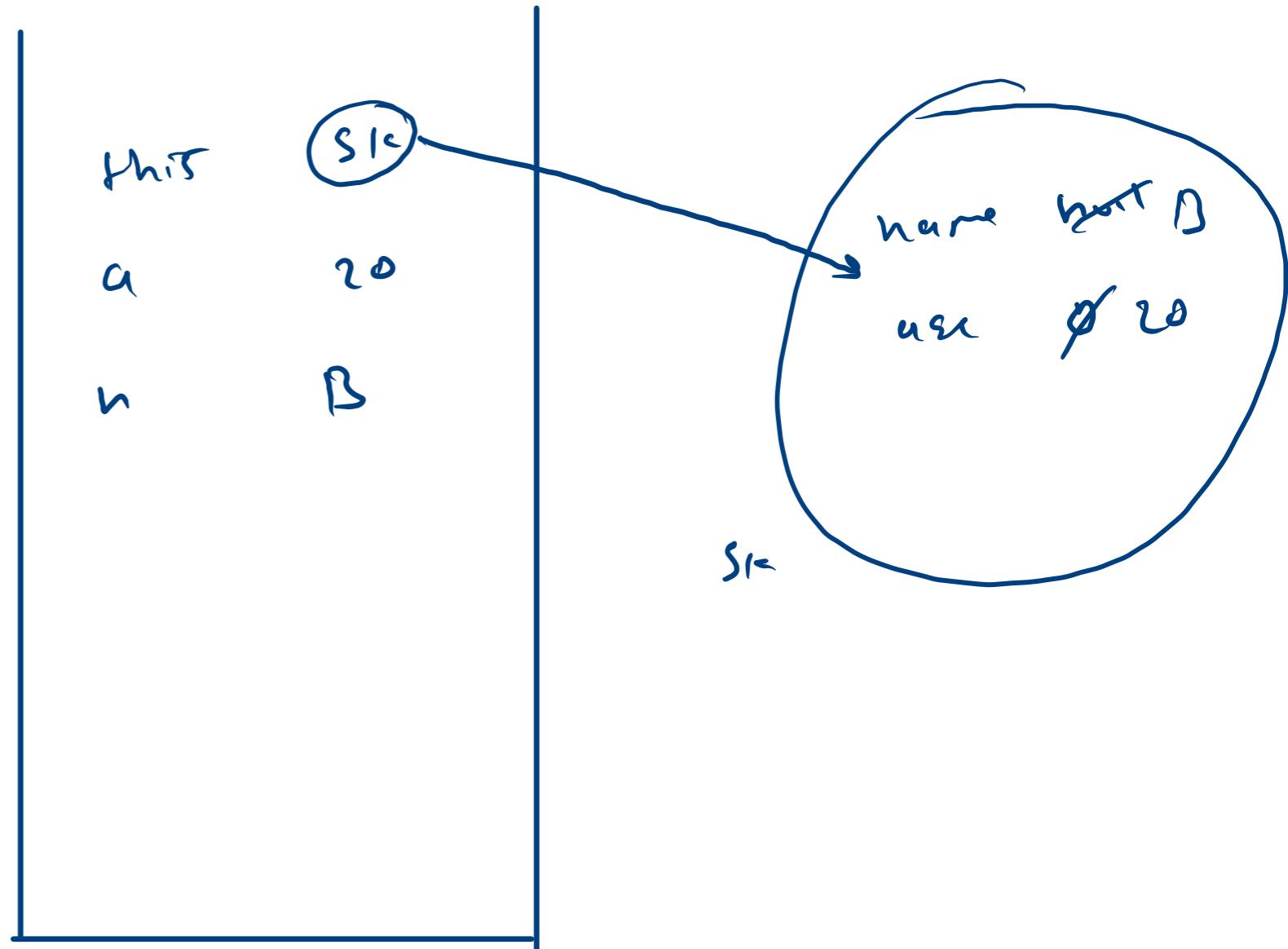
public void sayHii(){
    System.out.println(name+" "+ age +" says hii ");
}

public static void main(String[] args) {
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 10;

    p1.sayHii();

6 Person p2 = new Person("B", 20);
// p2.name = "B";
// p2.age = 20;
p2.sayHii();

```



push

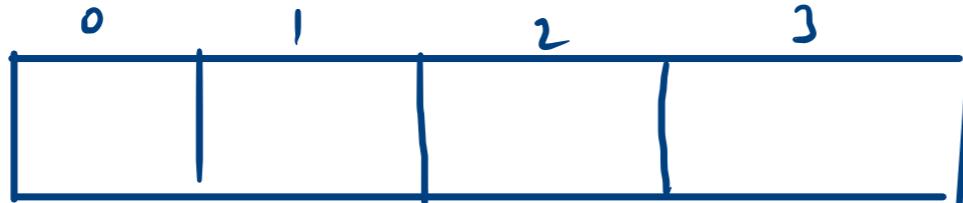
pop

peak → top

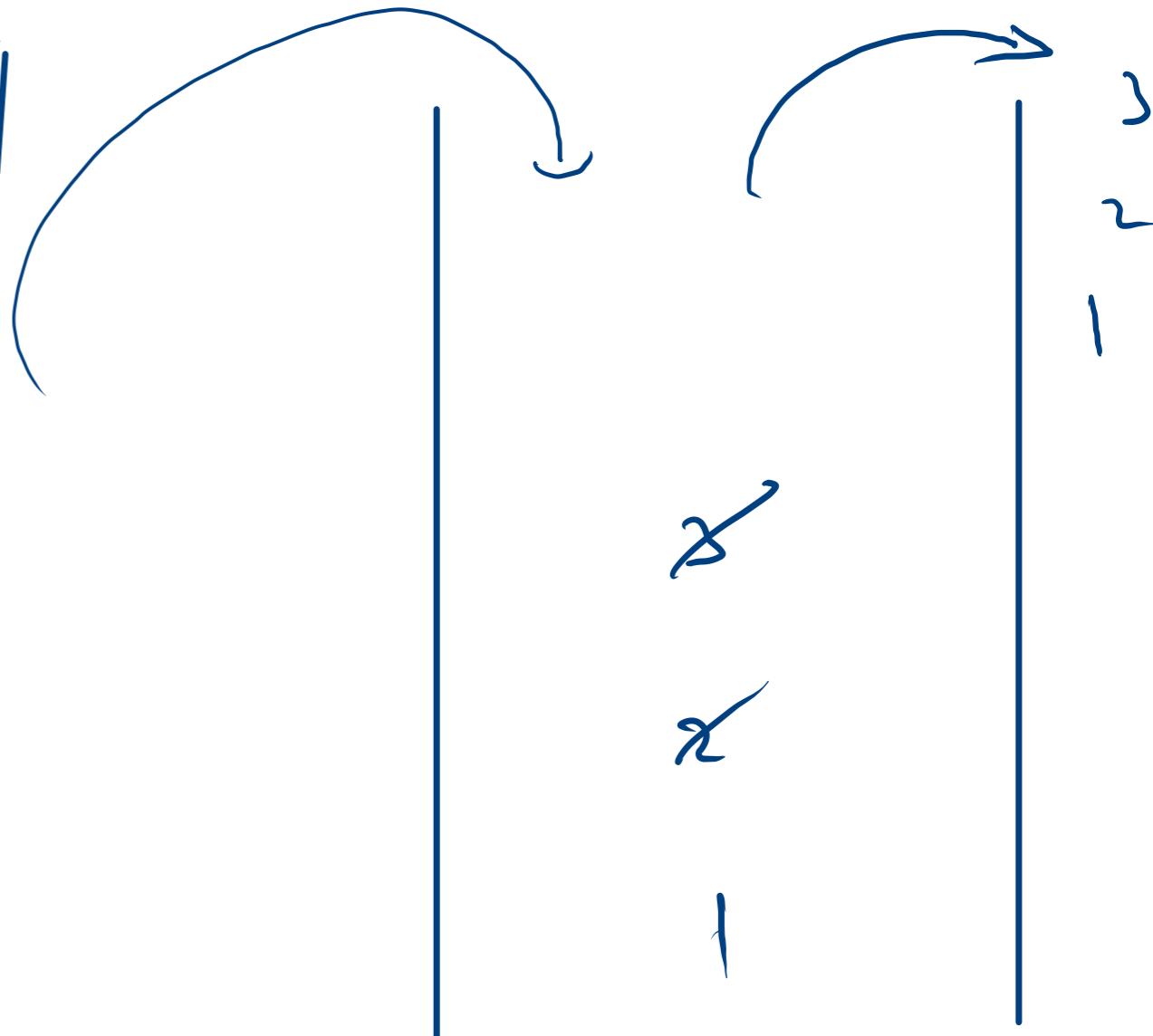
size →

display

to 5  
-1



cap = 4



~~5 push 10 display~~

~~push 20 display~~

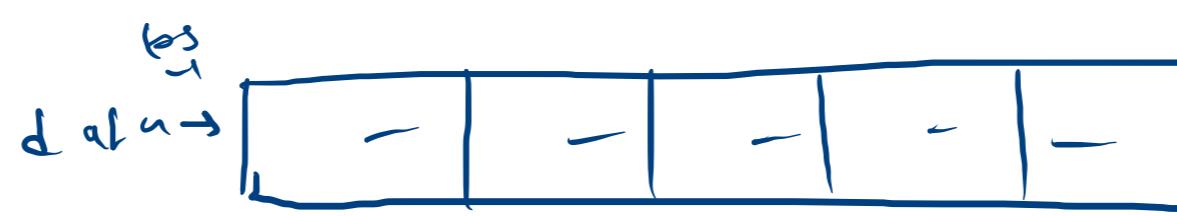
~~push 30 display~~

~~push 40 display~~

~~push 50 display~~

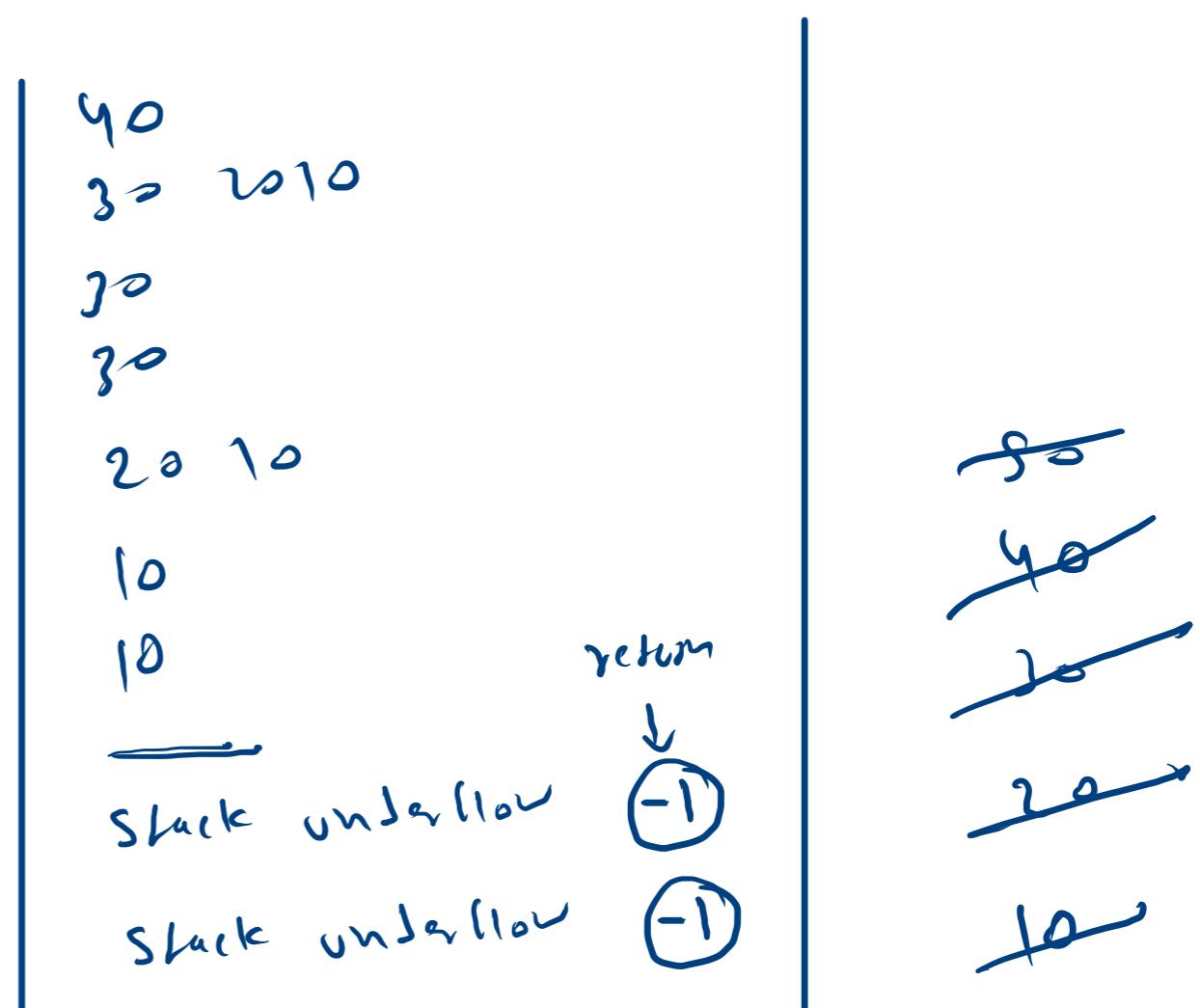
~~push 60 display~~

~~top pop quit~~

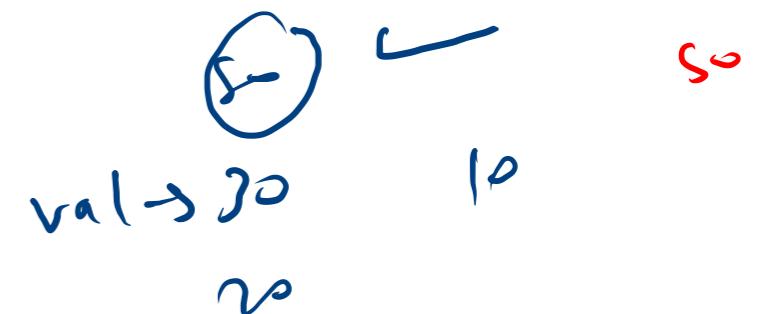
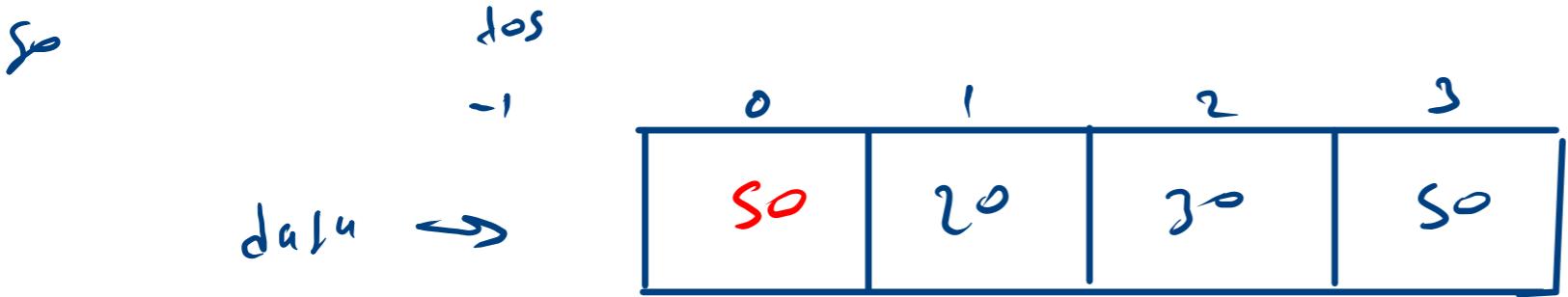


$$\text{Cap} = 5$$

10  
20 10  
30 2010  
40 302010  
50 40302010  
Stack overflow  
50 40302010  
80  
80  
40 2010  
40



push



pop

check for full

tos ++

data[tos] = val

check for empty

val = data[tos]

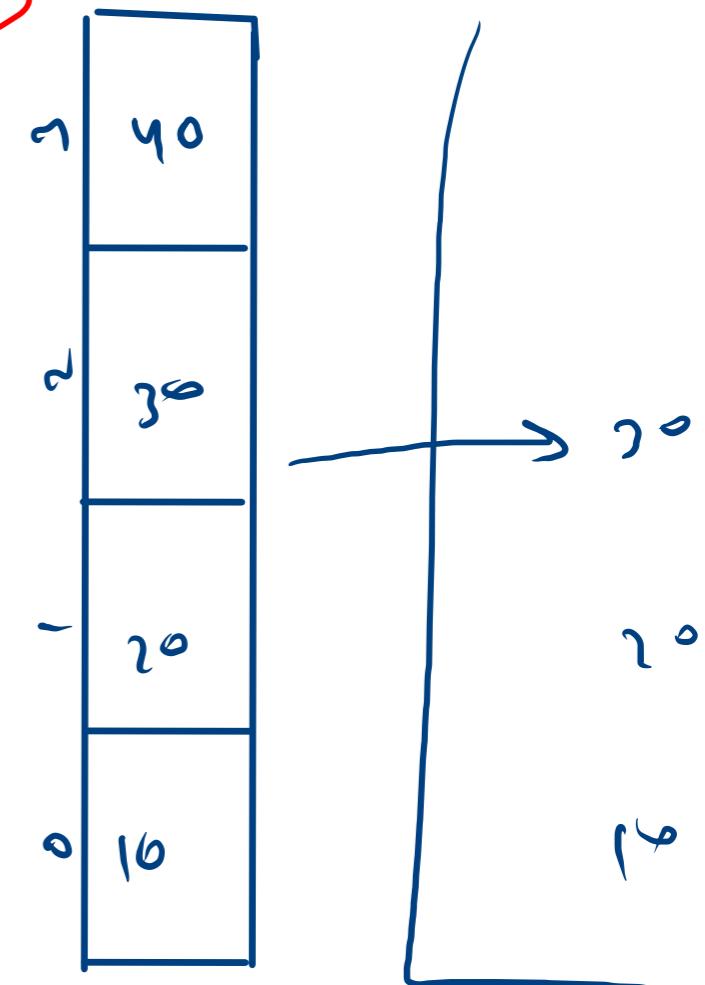
tos --  
return val;

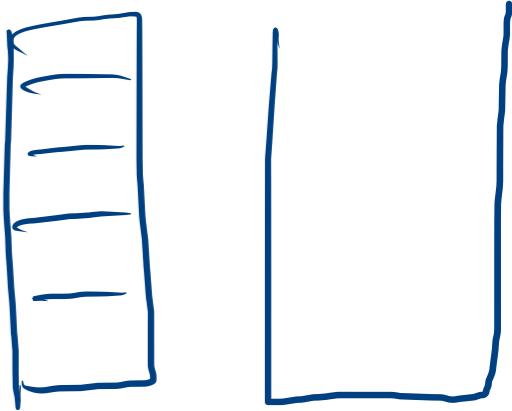
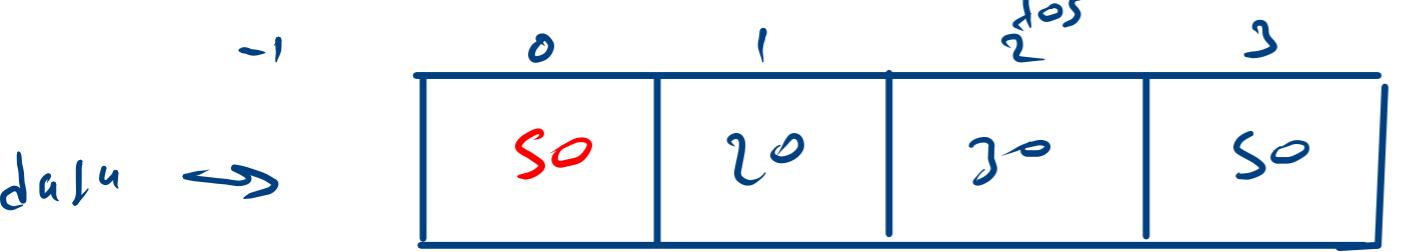
TOS

Checking for empty

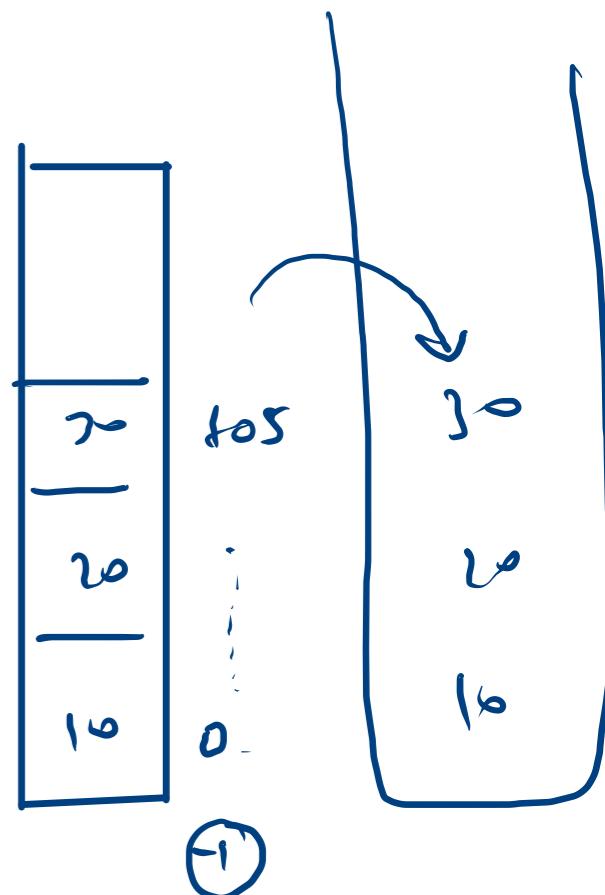
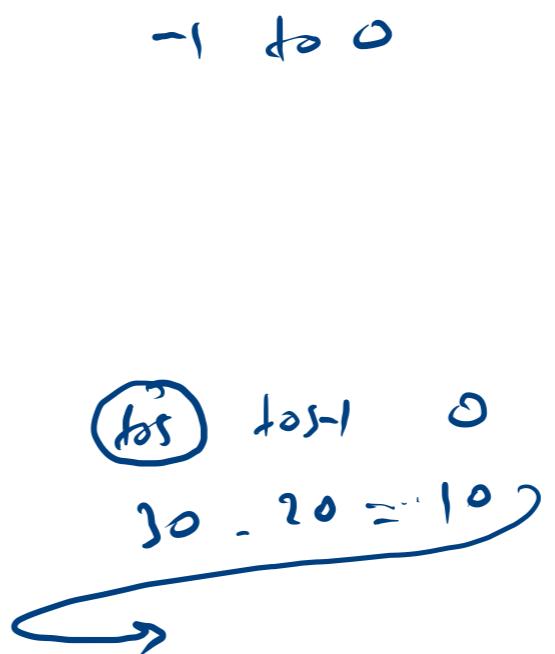
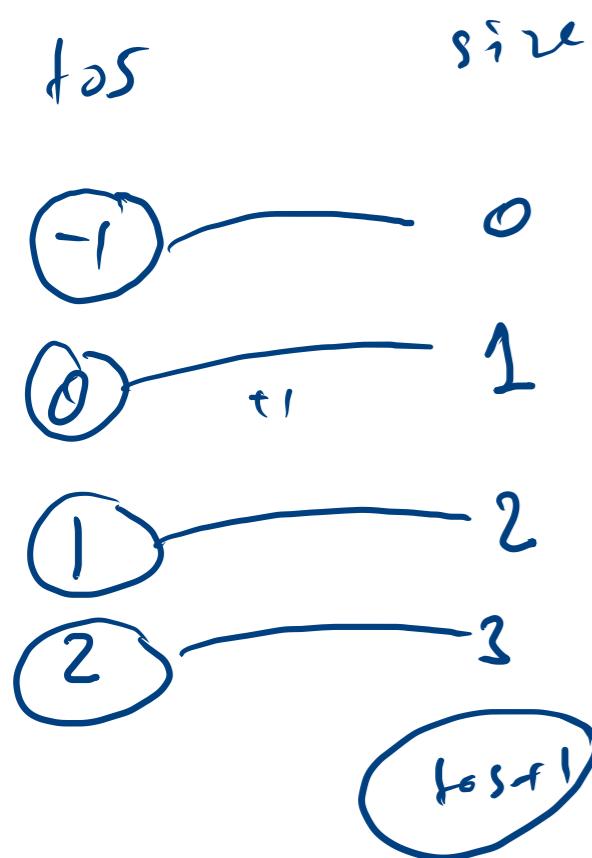
data[tos]

tos →

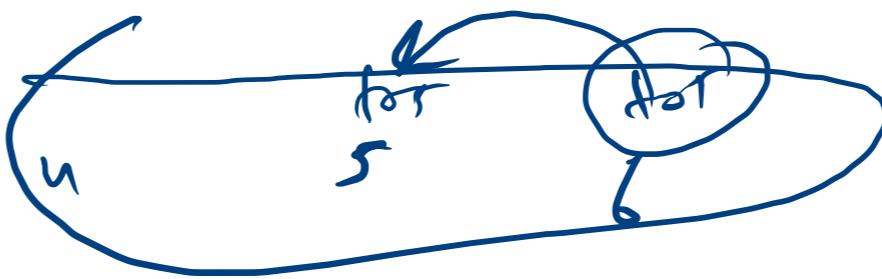




tos -1

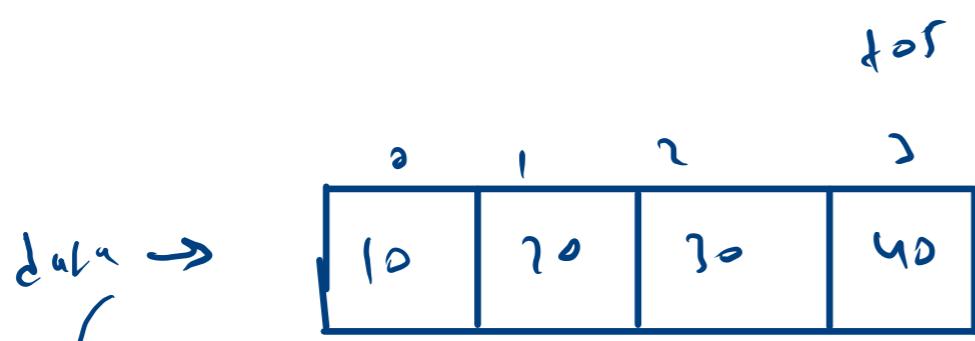


0	1	2	3
10	20	30	40



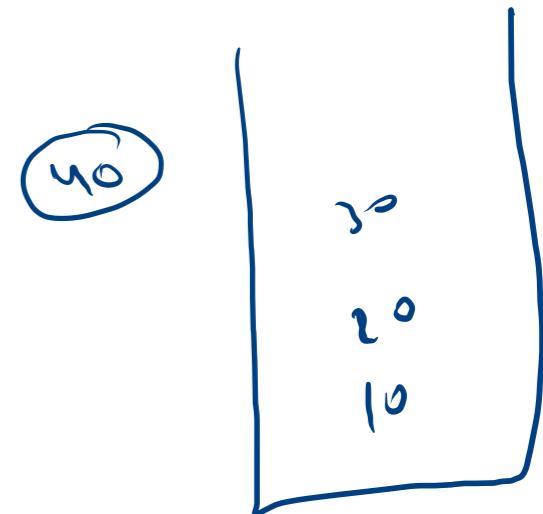
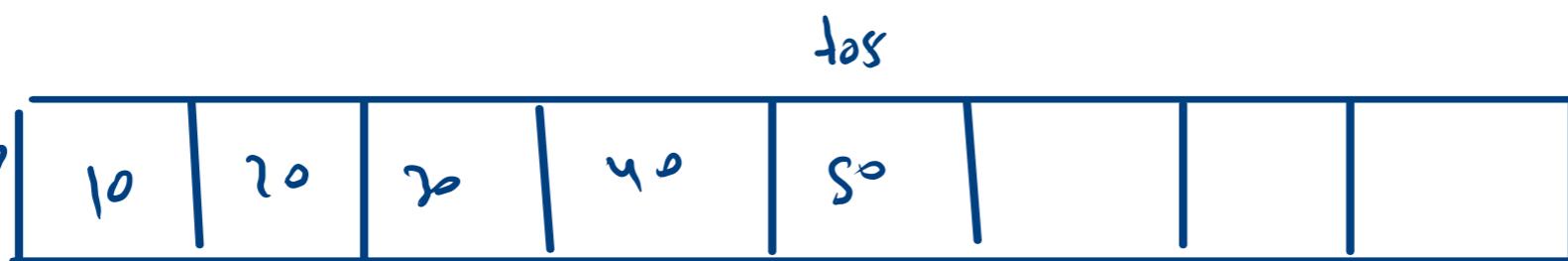
size 2

erm



*data →*

*new array  
(copy)*



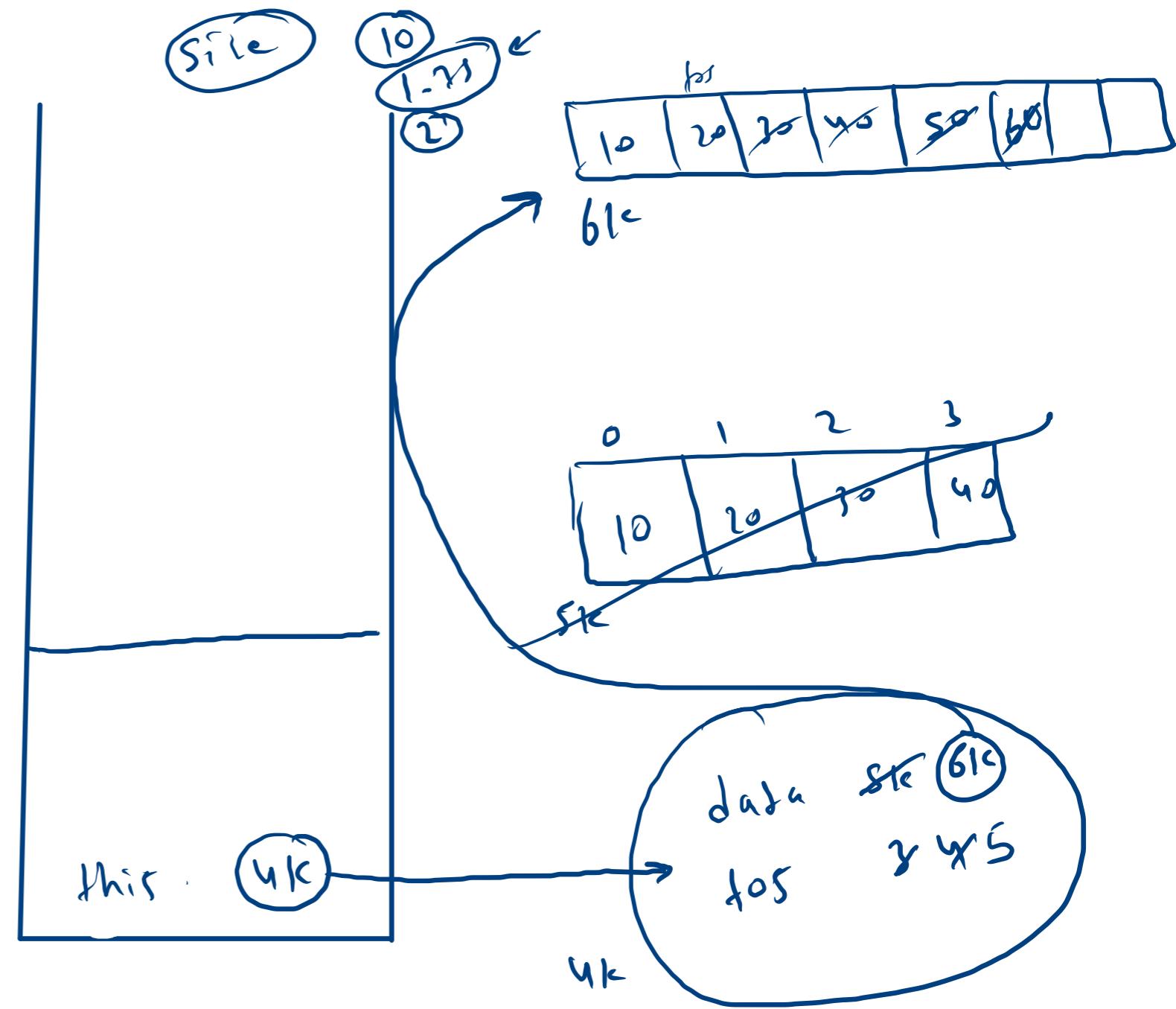
*ndata = new int(2 \* si)  
copy from data  
to ndata  
data = ndata*

```

void push(int val) {
    if (tos == data.length - 1) {
        int ndata[] = new int[data.length * 2];
        for (int i=0; i<data.length; i++) {
            ndata[i] = data[i];
        }
        data = ndata;
    }
    tos++;
    data[tos] = val;
} else {
    tos++;
    data[tos] = val;
}
}

```

push



~~push 10 push 20~~  
~~push 5 push 8~~  
~~push 2 push 4~~  
~~push 11 top min~~  
~~pop top min pop~~  
~~top min pop top~~  
~~min pop top min~~  
~~pop top min pop~~  
~~top min pop quit~~

11  
2 min  
11  
4 2 ← min  
4 2  
2  
2 ← min  
2 8  
5 ← min  
8  
5  
5 → min

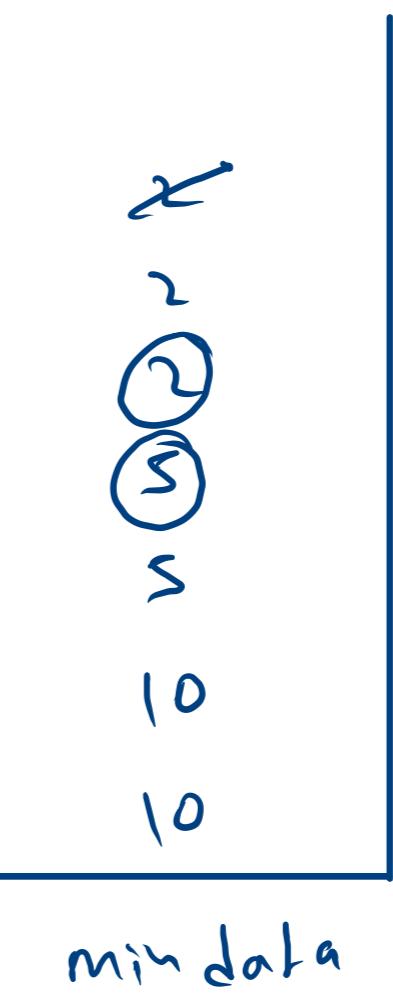
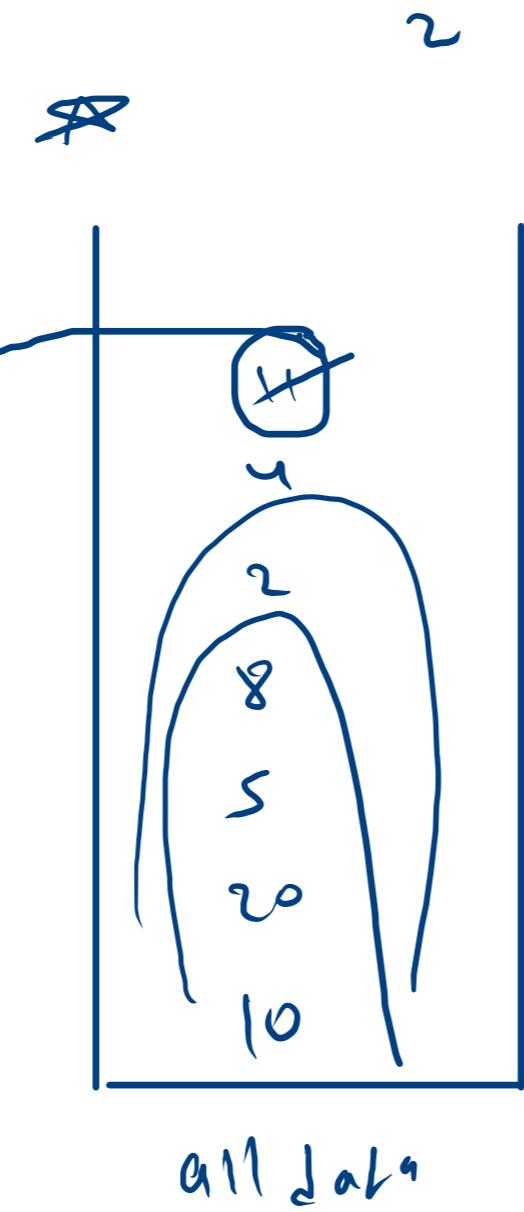
5  
20  
10 → min  
20  
10  
10 ← min  
10

x  
x  
2  
x  
8  
8  
x  
x  
10

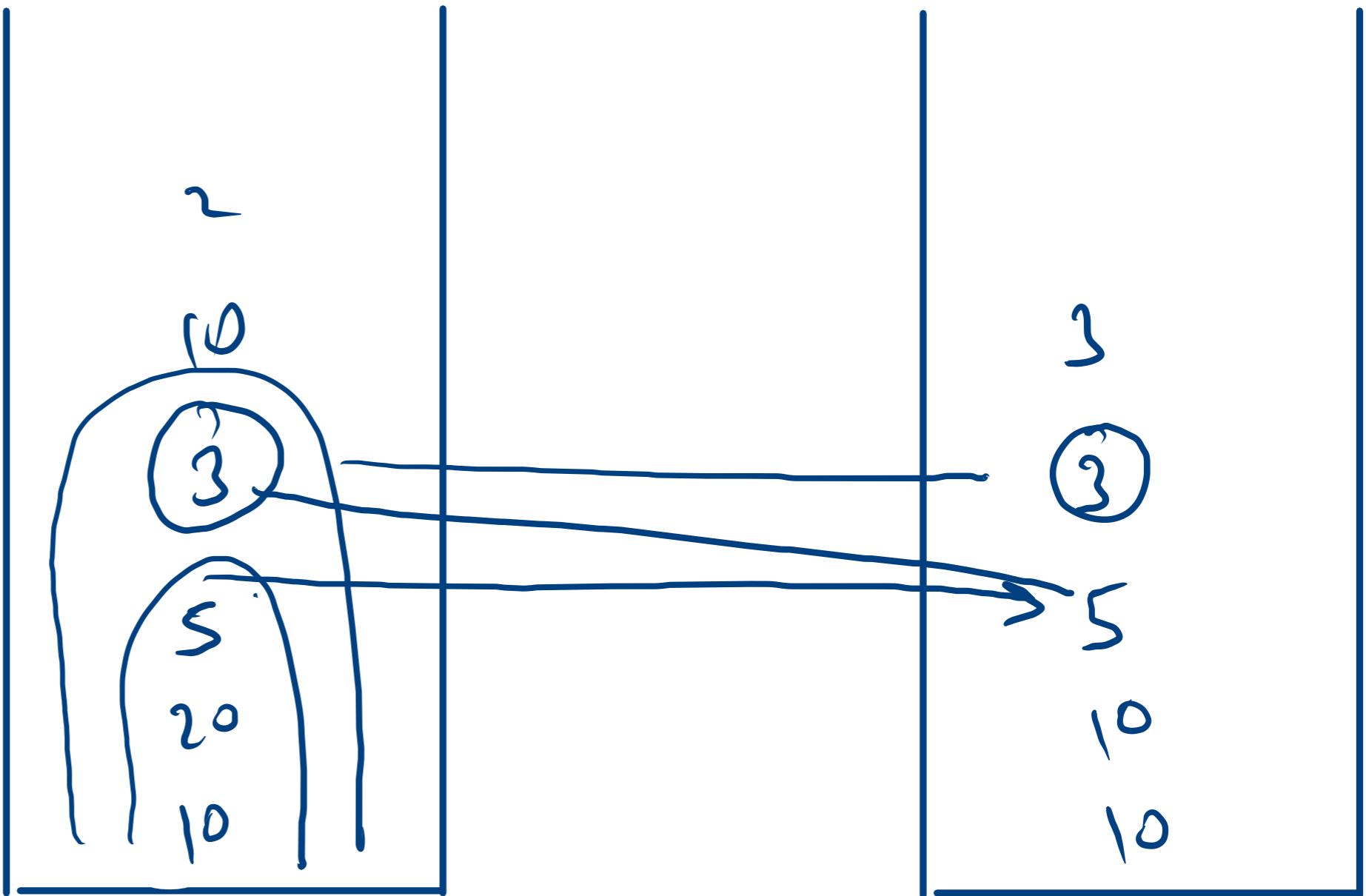
push  
pop  
for  
size  
+ min

~~push 10 push 20~~  
~~push 5 push 8~~  
~~push 2 push 4~~  
~~push 11 top min~~  
~~pop top min pop~~  
top min pop top  
min pop top min  
pop top min pop  
top min pop quit

11  
2  
11  
4



push all data min



$$n = \text{peak} + \text{val} - \text{min}$$

$$\text{peak} = \text{min} + \text{min} - \text{peak}$$

$$\text{min} = \text{min} + \text{min} - \text{peak}$$

$$2+2 - (-1)$$

$$2+2 + 1 = 5$$

$\text{val} = 5$

$\text{val} < \text{min}$

$n < \text{val}$

$n = \text{val}$

$n = s + -s$

$\approx 0$

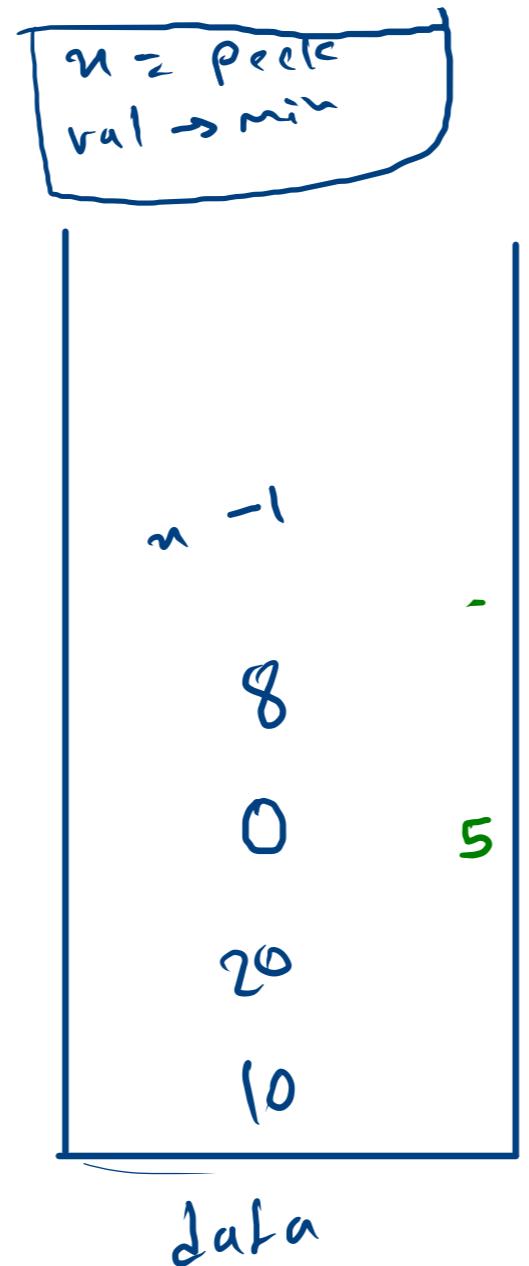
$\text{val} - \text{min}$

$s - 10$

$-5$

$s$

$10$



$n = \text{val} + \text{val} - \text{min}$

$\text{peak} = \text{min} + \text{min} - \text{peak}$

$\text{min} = \text{min} + \text{min} - \text{peak}$

$2+2 - (-1)$

$2+2 + 1 = 5$

$\leftarrow 2 \text{ min}$

$\leftarrow \text{min}$

$5 < \text{min}$

$5$

$10$

$10$

$\text{min}$

$\text{min} \rightarrow 5$

$$\begin{aligned}
 n &= val + val-min \\
 5 &+ 5 - 10 \\
 &\approx 5 - 5 \\
 &\approx 0
 \end{aligned}$$

$$val > min$$

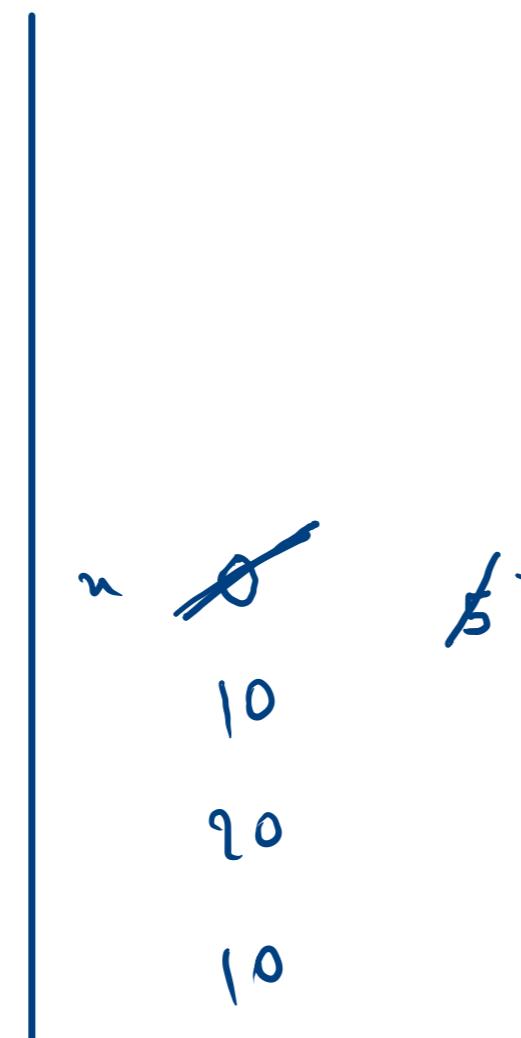
$$20 > 10$$

$$10 = 10$$

$$5 < 10$$

min charn

Pop



$$\begin{aligned}
 n &= val + val-min \\
 5 &+ 5 - 10 \\
 &\approx 5 - 5 \\
 &\approx 0
 \end{aligned}$$

$$\begin{aligned}
 peek &= min + min - peek \\
 min &= min + min - 0 \\
 &\approx 10 \\
 min &> 8 \quad 10
 \end{aligned}$$

$$n = val + val-min$$

$$= 2 + 2 - 3$$

$$= 1$$

$$val > min$$

$$20 > 10$$

$$10 = 10$$

$s < 10$



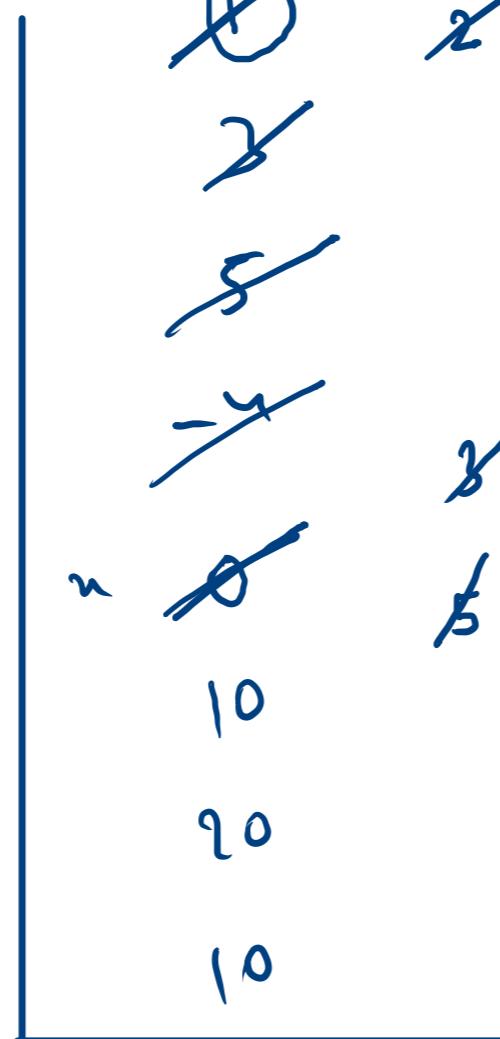
min channel

$$3 < 10$$

$$5 > 3$$

$$2 = 3$$

$$2 < 3$$



$$\frac{1}{2} < \frac{2}{3} = \frac{2}{3}$$

$$\frac{5}{4} > \frac{3}{2}$$

$$\frac{-4}{3} < \frac{3}{2}$$

$$\frac{5}{4} > \frac{3}{2}$$

$$\frac{-4}{3} < \frac{3}{2}$$

$$p_{min} = min + min - peak$$

$$3 + 3 - (-4)$$

$$3 + 3 + 4$$

$$10$$

min  $\rightarrow$  8 10 8  
 $\leftarrow$  8 10