

$$a + b \neq c$$

for

$$a + bc \neq$$

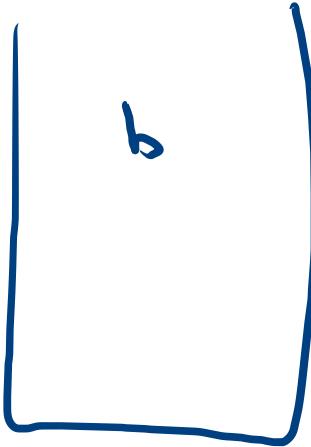
$$a b c \neq f$$

pre

$$a + \star b c$$

$$+ a \neq b c$$

v_2
 v_1



I co-r

4
0000
1011
1101
1110

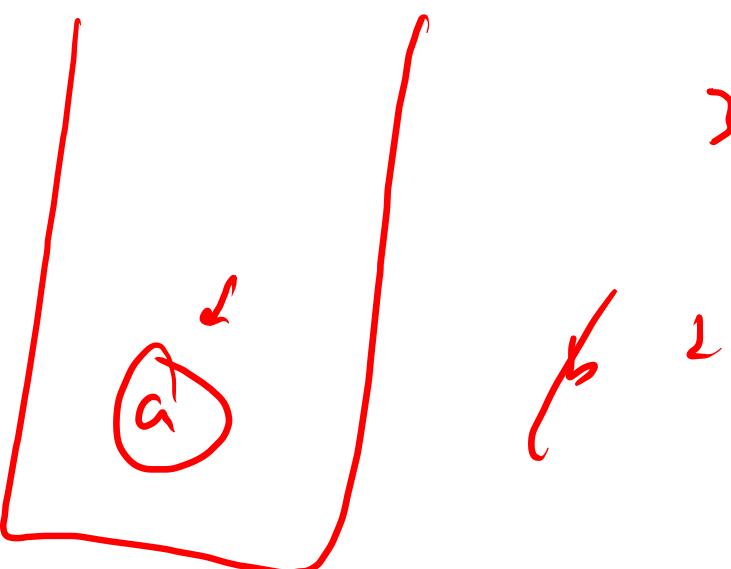
	a	b	c	d
0	a	0	0	0
1	b	1	0	1
2	c	1	1	1
3	d	1	1	0

hex

hex

	a	b	c	d
0	1	0	1	0

	a	b	c	d
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0

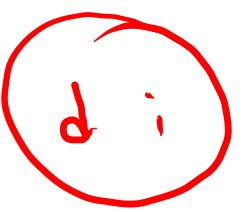


a d

b c

a

52



d d

f

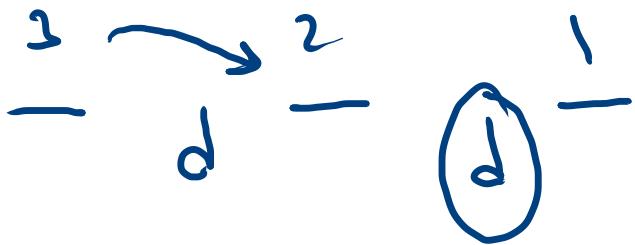
g

g

1

2

3



1...9

d -> 21

i -> 12

ddd -> 4321

iii -> 1234

ddidd -> 43218765

iiddd -> 126543

4 - 2 - 3 - 2 - 1 - 8 - 7 - 6 - 5 -

3 - 2 - 1 - ; - 7 - 6 - 5 - 4 - ; -

3 2 1 4 6 5 7 8

3 2 1 4 6 5 7 8

3 2 1 4 6 5 7 8

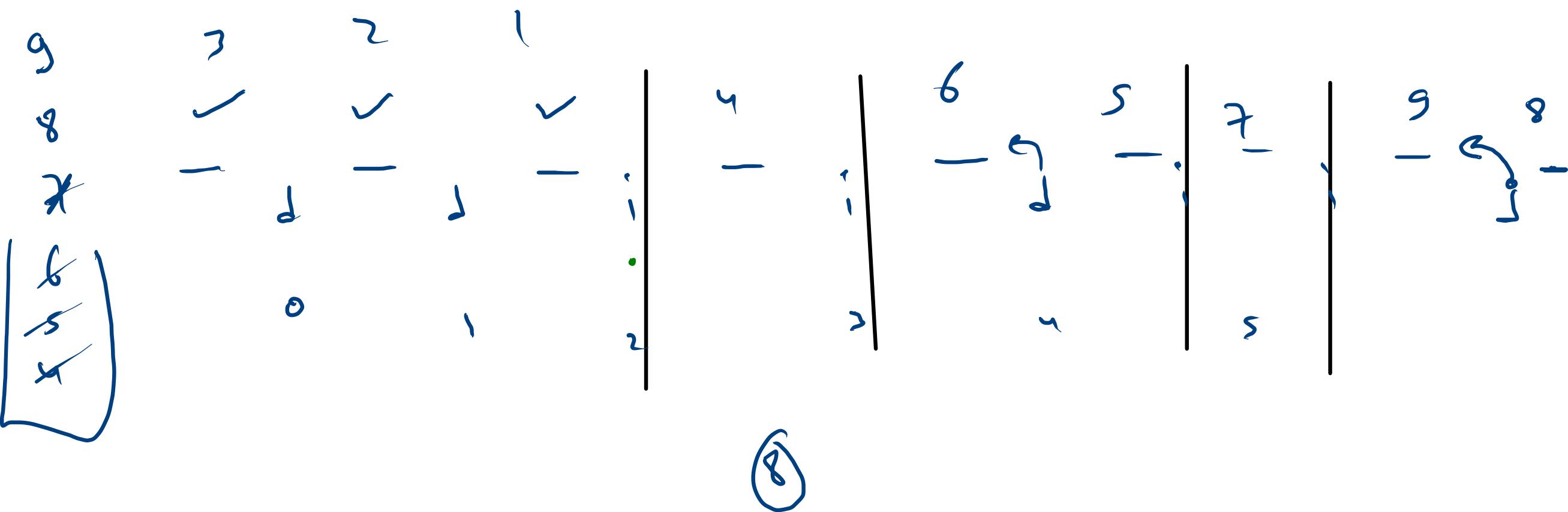
3 2 1 4 6 5 7 8

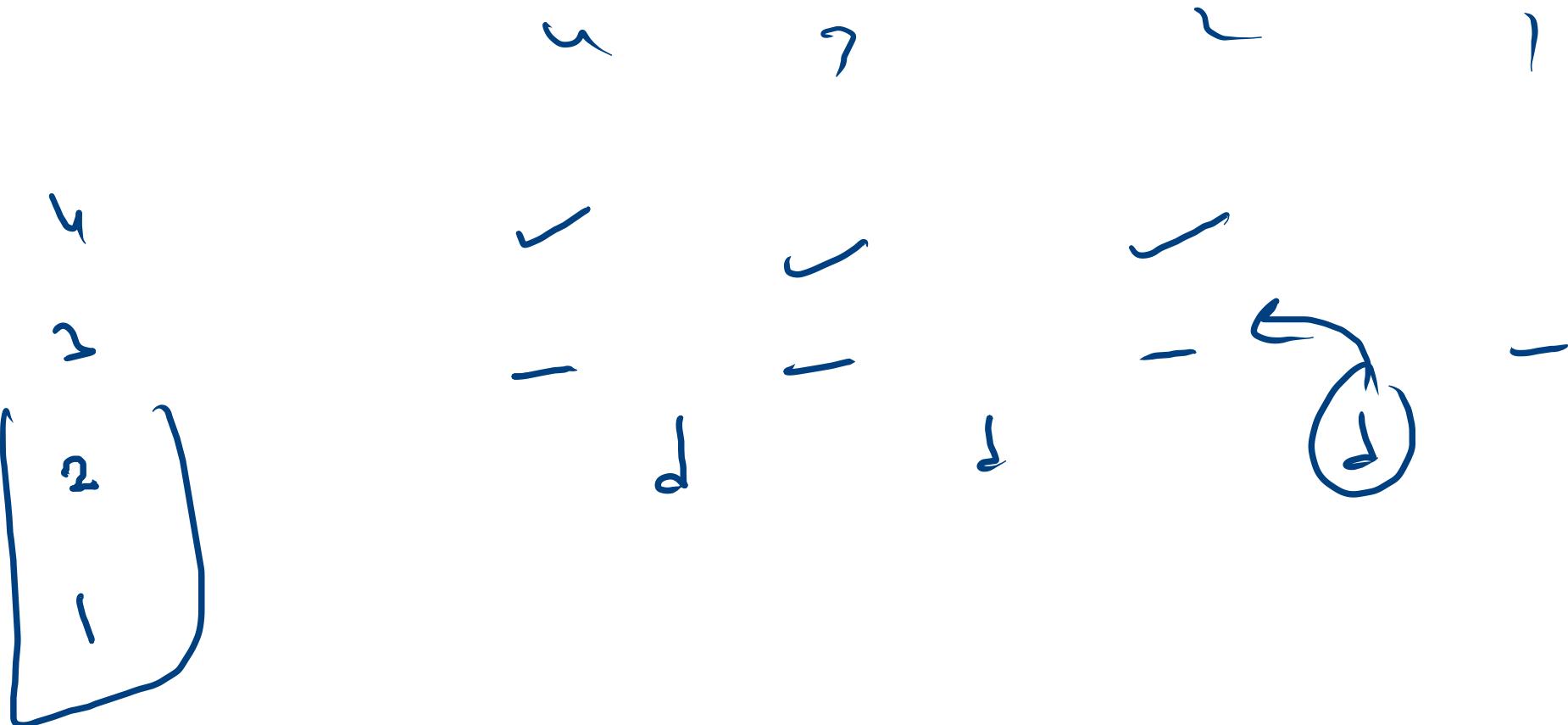
3 2 1 ; 4 ; 5 ; 6 ; 7 ; 8

ddi i di

val

x < 84567



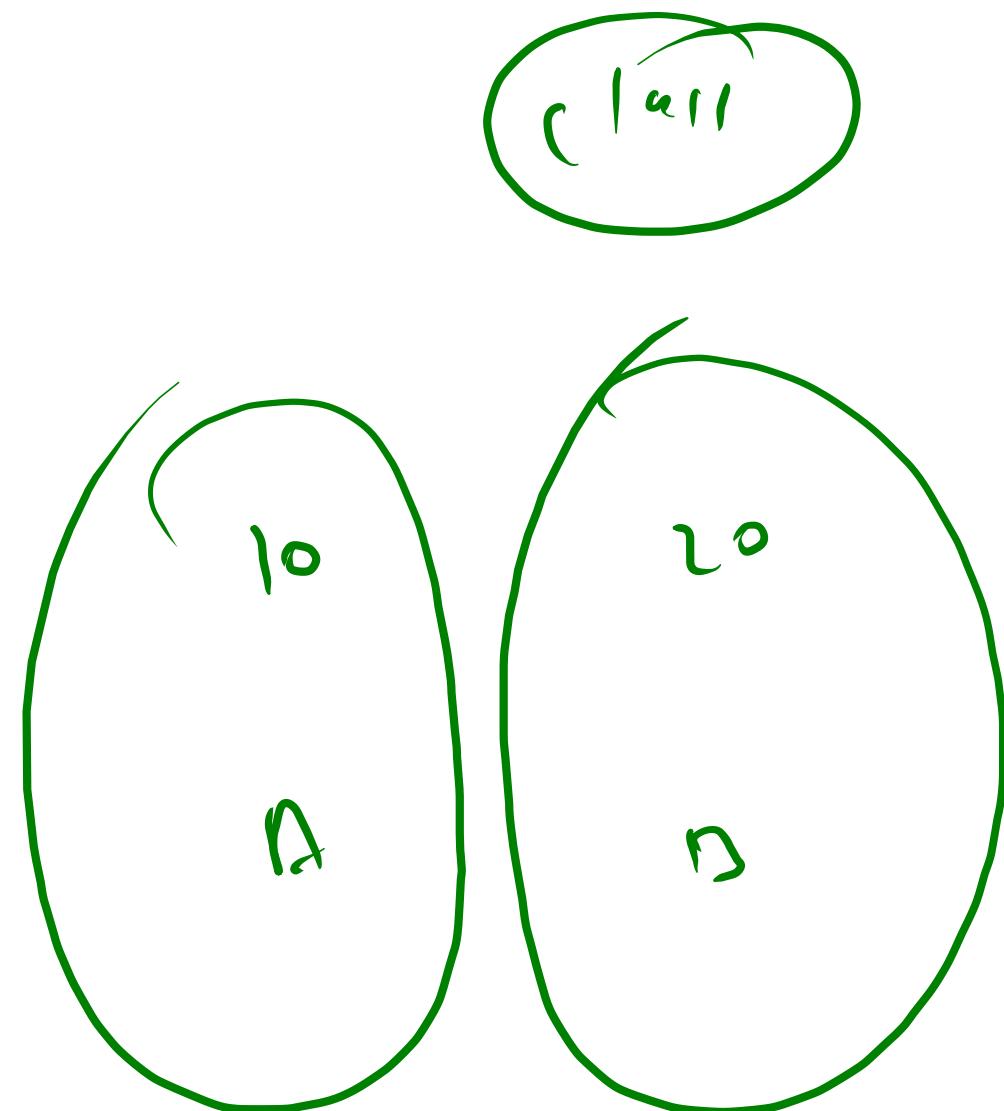
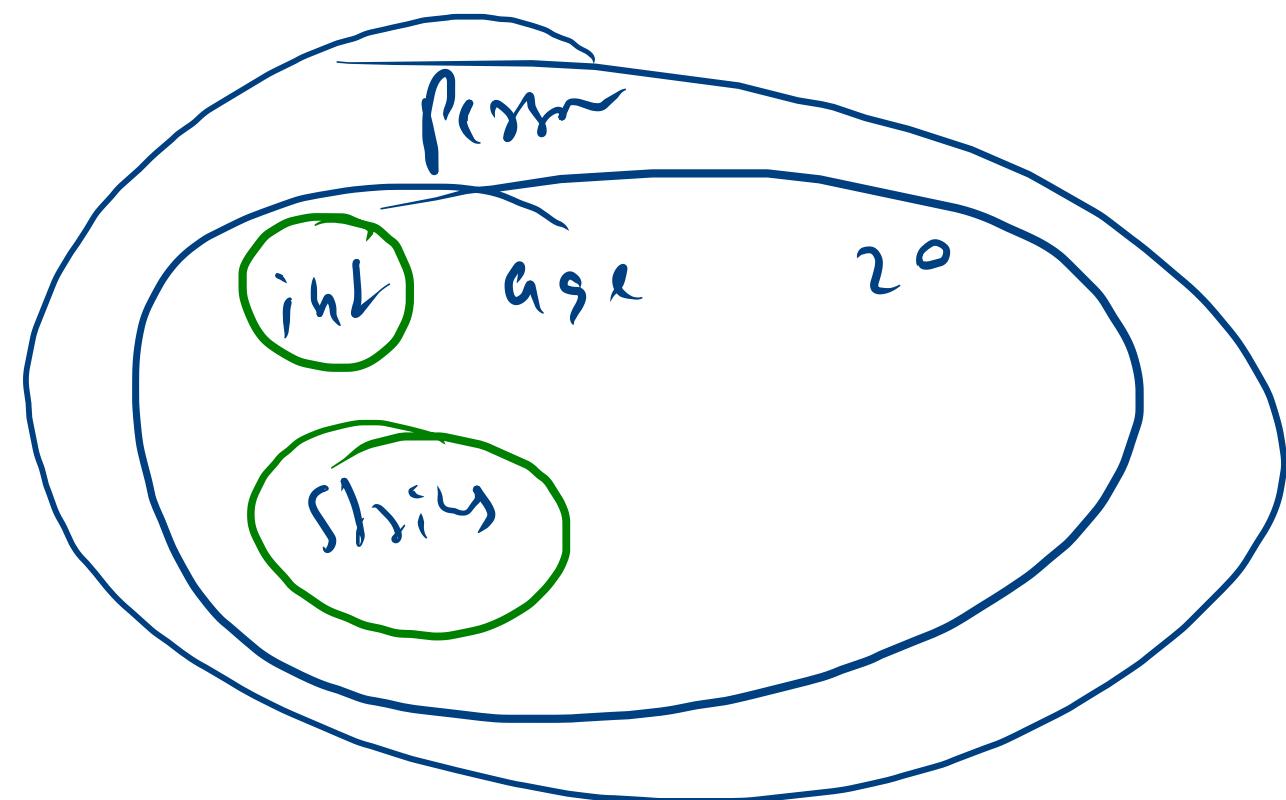


array

arraylist

ihl

(- - -)



class

```

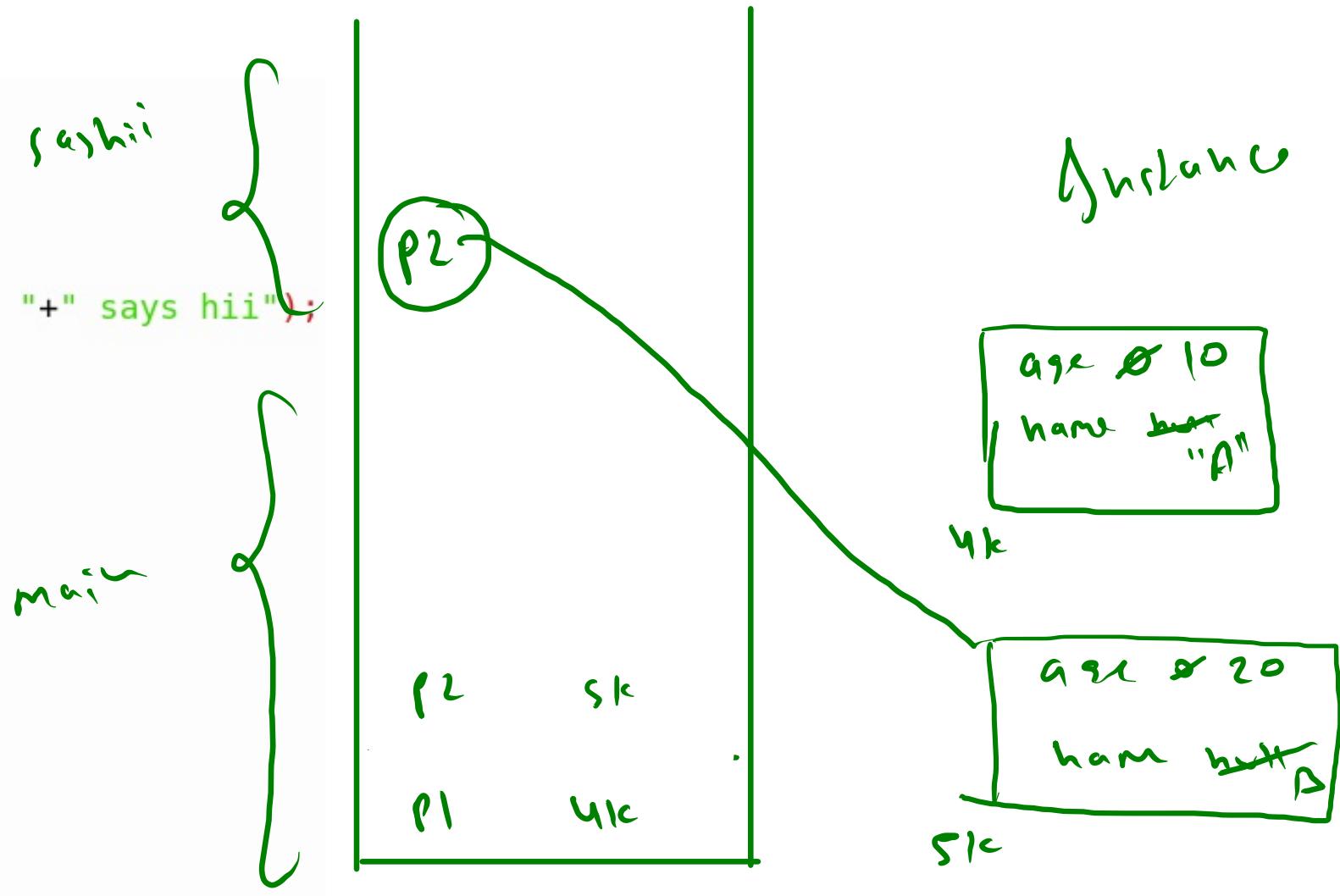
static class Person{
    int age;
    String name;

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

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

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

```



A 10 sashi

B 20 sashi

```

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

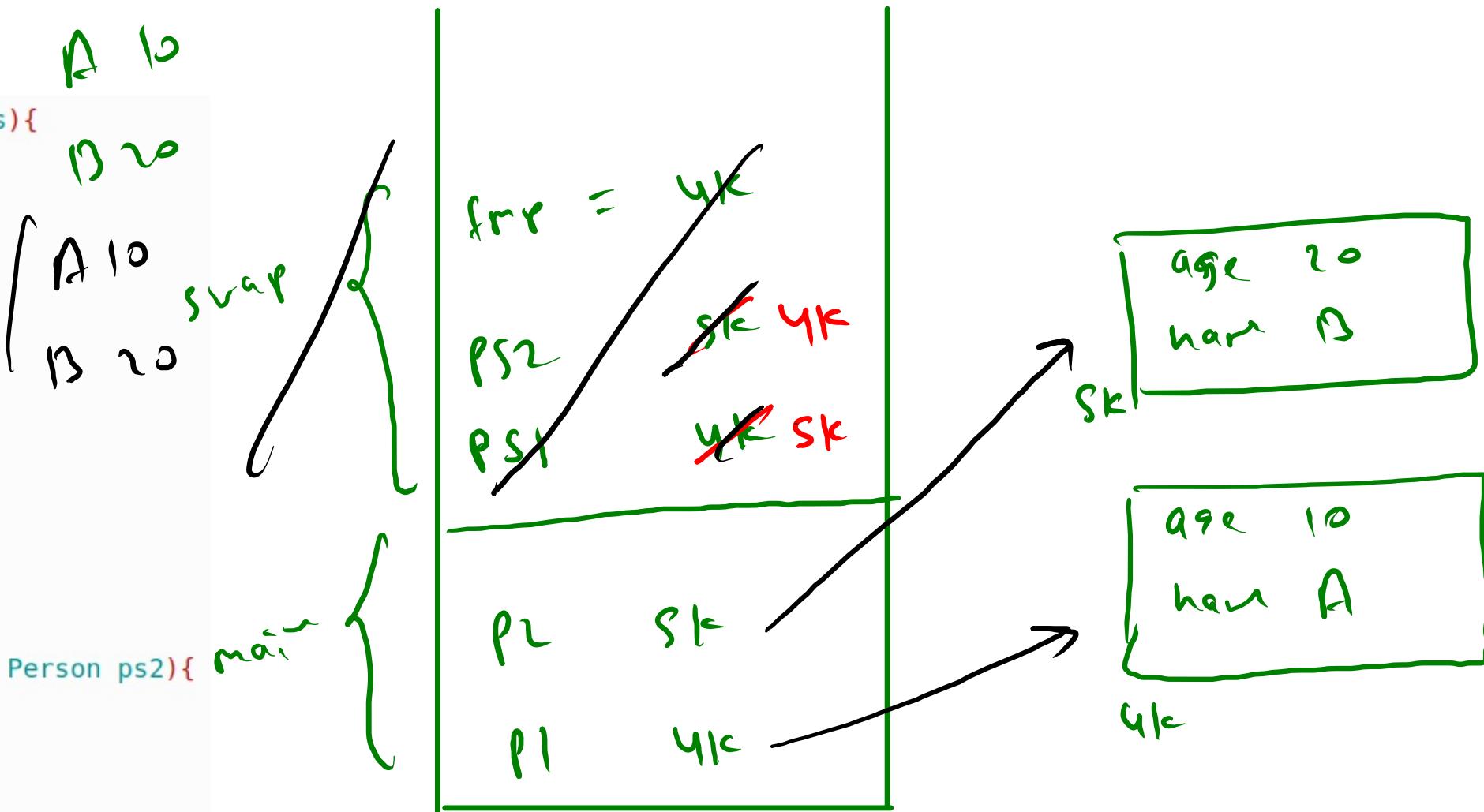
    Person p2 = new Person();
    p2.age = 20;
    p2.name = "B";
    p2.sayHii();

    swap(p1, p2);

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

public static void swap(Person ps1, Person ps2){
    Person tmp = ps1;
    ps1 = ps2;
    ps2 = tmp;
}

```



```

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

// swap1(p1, p2);
swap2(p1, p2);

p1.sayHii(); ✓
p2.sayHii(); |
```

```

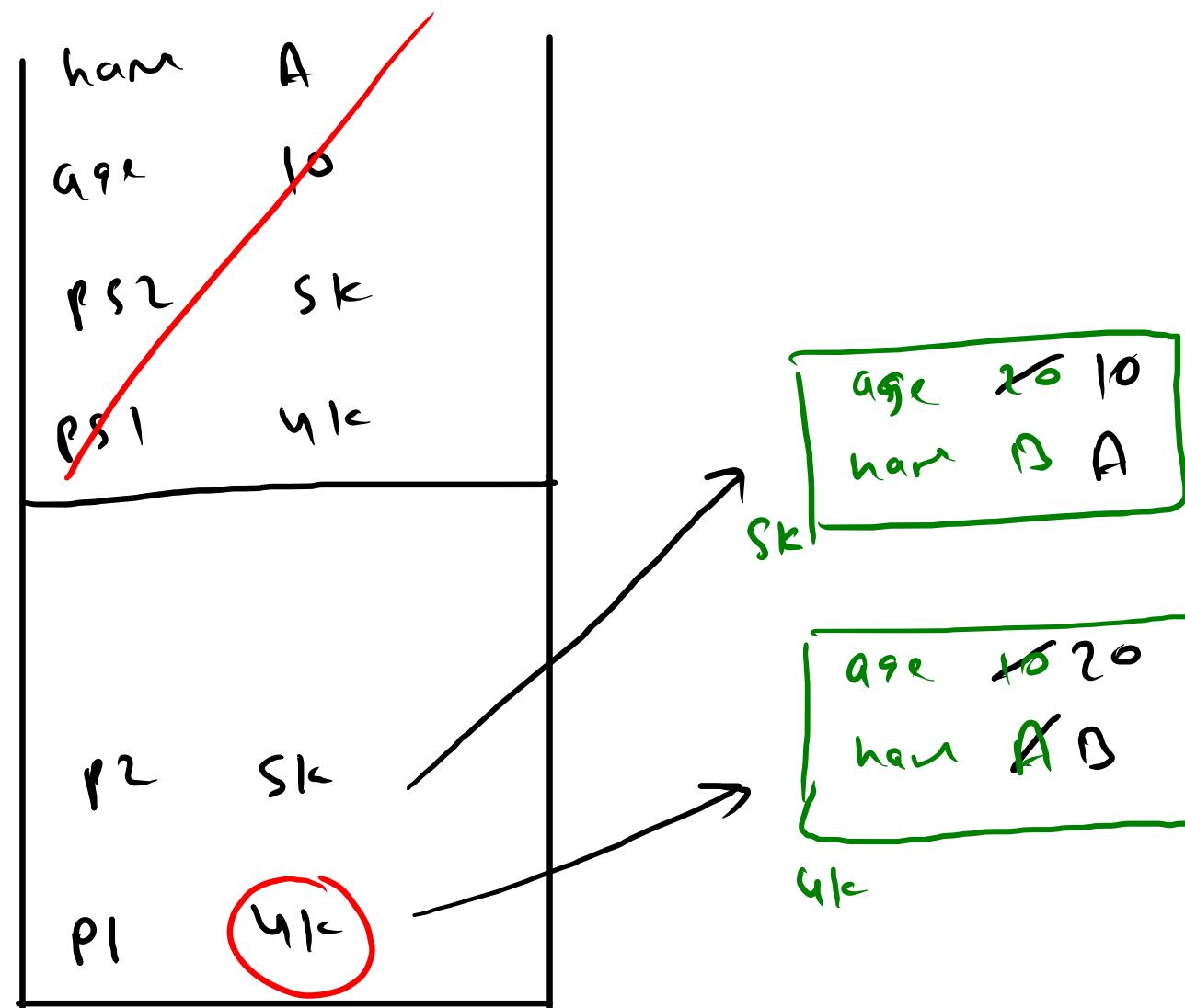
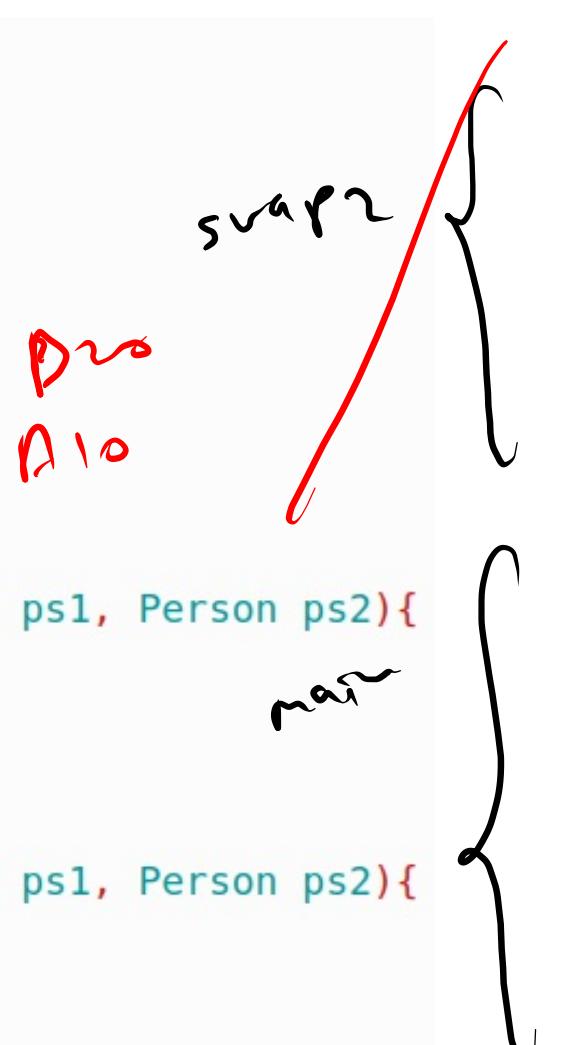
public static void swap1(Person ps1, Person ps2){
    Person tmp = ps1;
    ps1 = ps2;
    ps2 = tmp;
}
```

```

public static void swap2(Person ps1, Person ps2){
    int age = ps1.age;
    ps1.age = ps2.age;
    ps2.age = age;
```

```

    String name = ps1.name;
    ps1.name = ps2.name;
    ps2.name = name;
```



```

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

// swap1(p1, p2);
// swap2(p1, p2);
swap3(p1, p2);

p1.sayHii();
p2.sayHii();

}

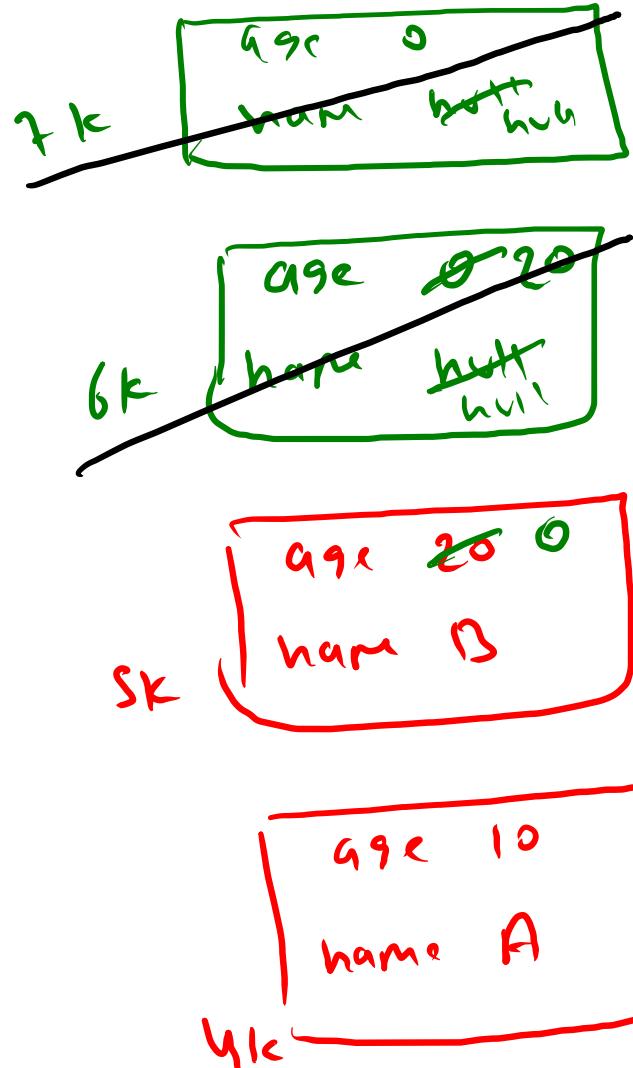
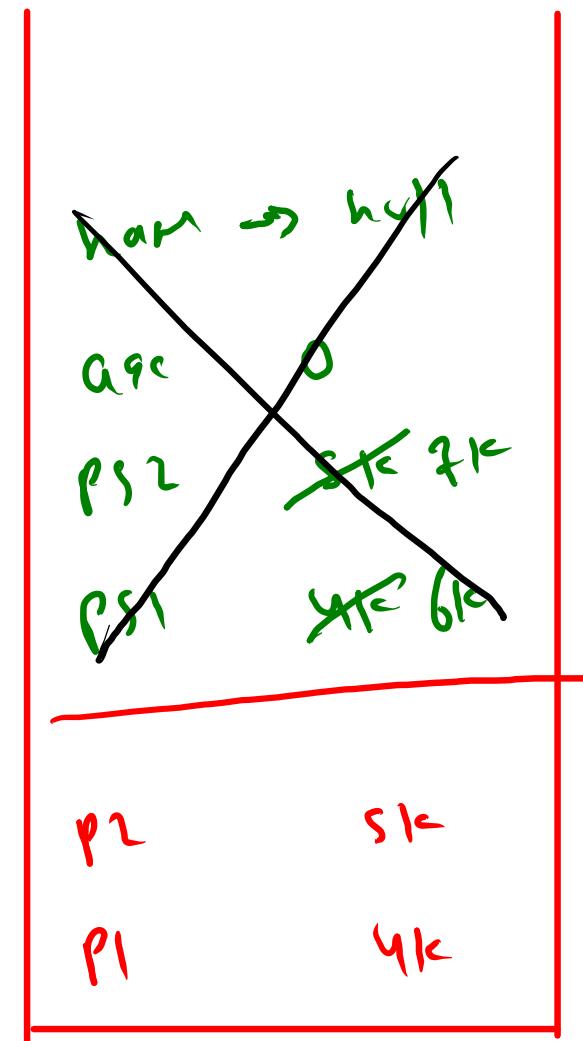
public static void swap3(Person ps1, Person ps2){
    ps1 = new Person(); ✓

    int age = ps1.age; ✓
    ps1.age = ps2.age; ✓
    ps2.age = age; ✓

    ps2 = new Person(); ✓
    String name = ps1.name; ✓
    ps1.name = ps2.name; ✓
    ps2.name = name; ✓
}

```

p_1
 p_2
 Swap3
 $A \ 10$
 $B \ 0$



```

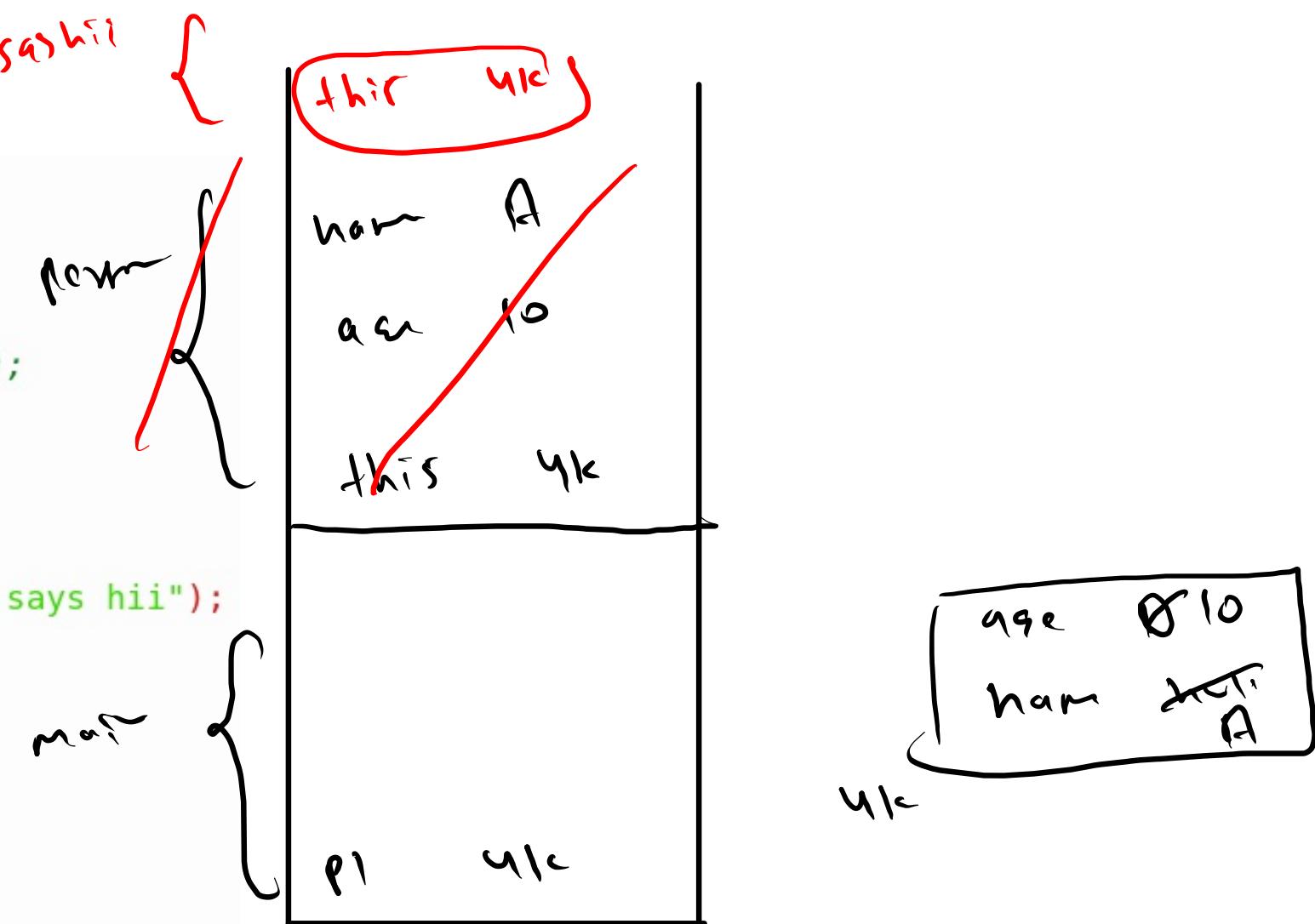
10 A
int age;
String name;

// default constructor
public Person(int age, String name){
    // System.out.println("created person");
    this.age = age;
    this.name = name;
}

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

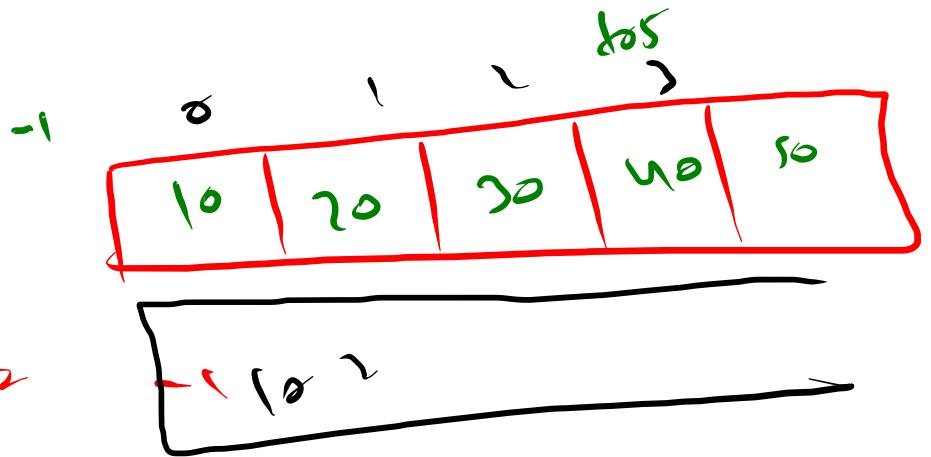
public static void main(String[]args){
    Person p1 = new Person(10, "A");
    // p1.age = 10;
    // p1.name = "A";
    p1.sayHii();
}

```

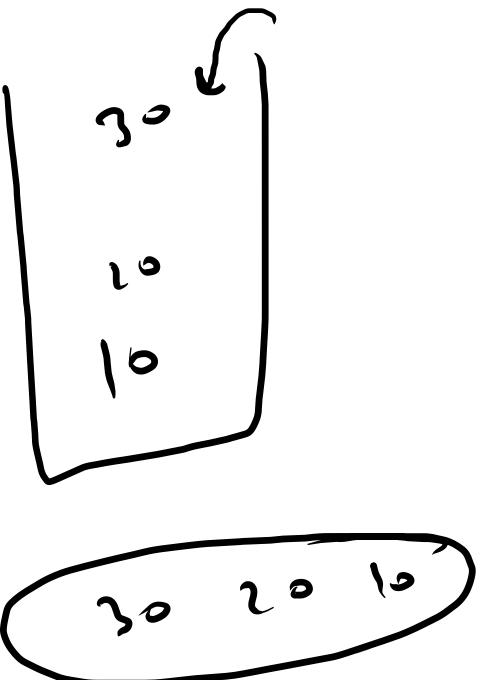


cap = 5

data →
→ **tos**
tos or slack



tos



push

push

10

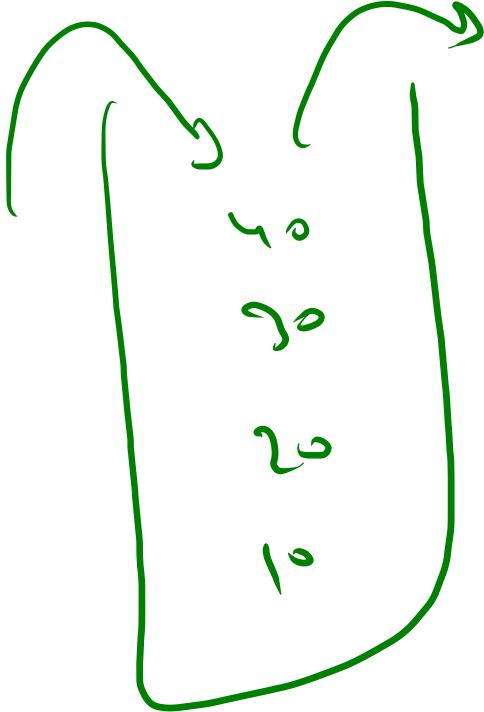
20

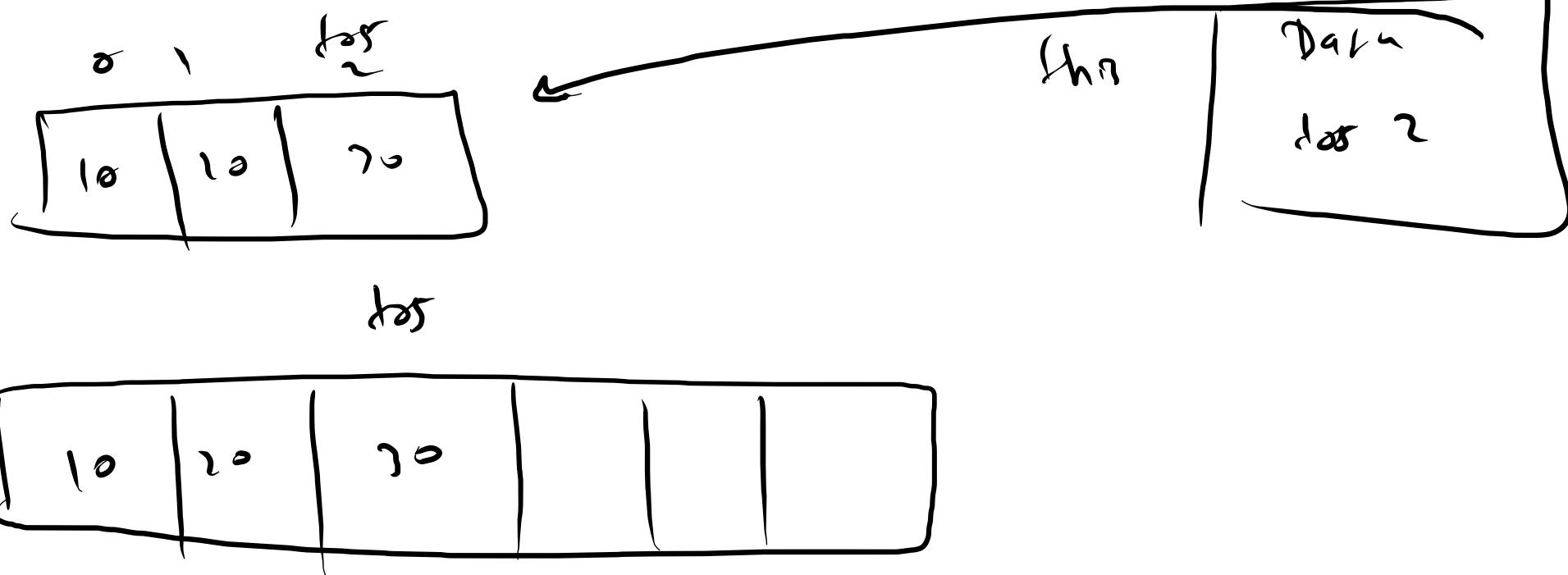
pop

pop

3

4





loc arr

Duh - her arr

arr

{

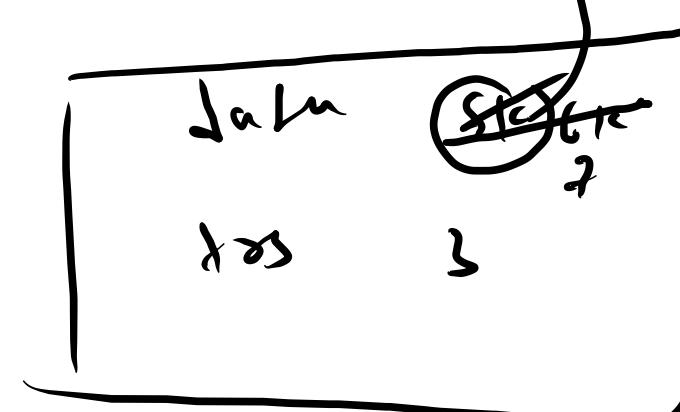
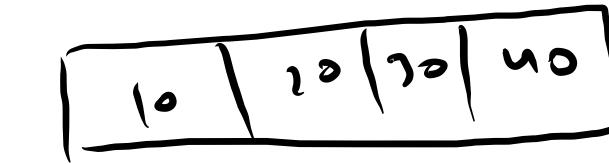
arr

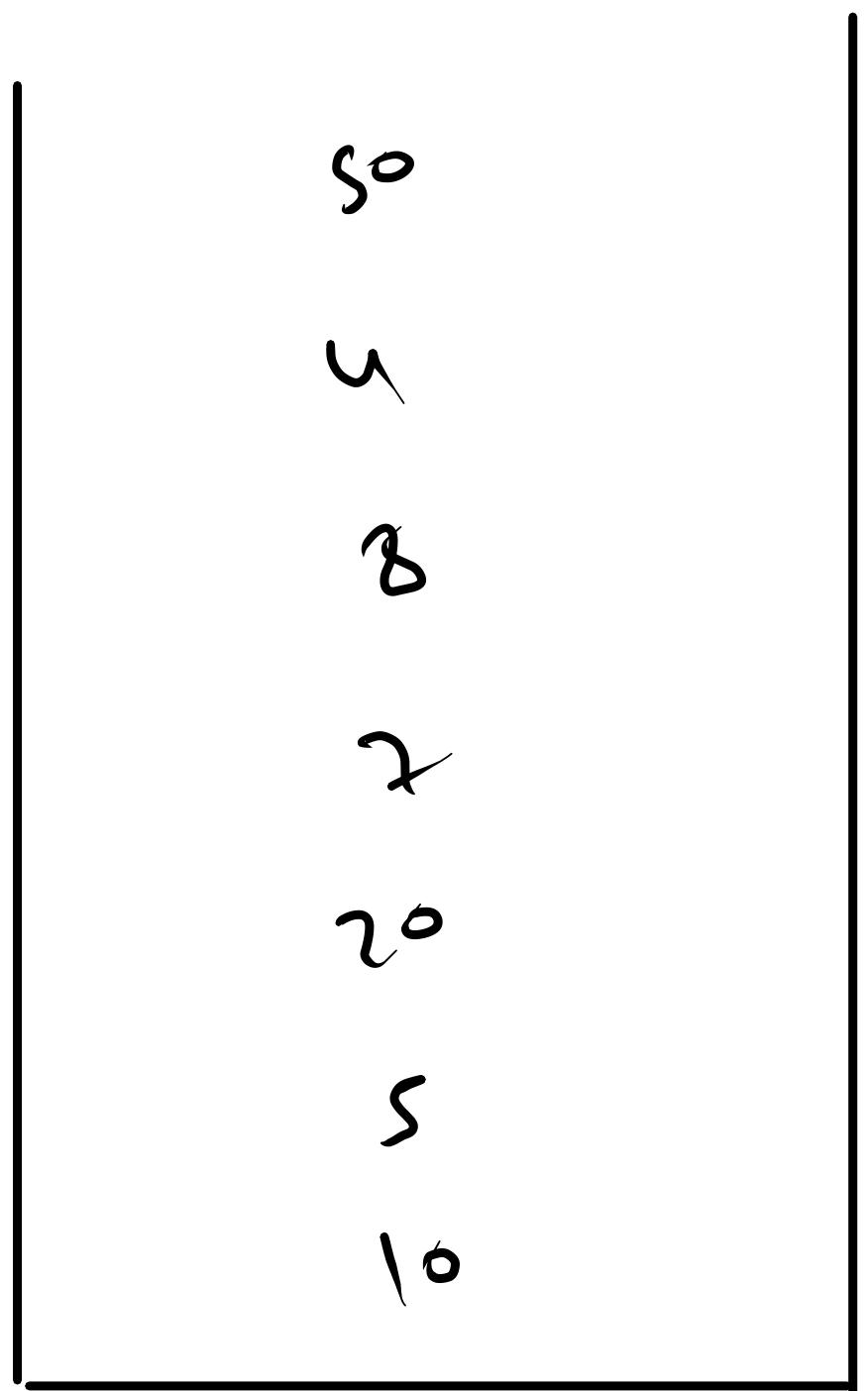
slc

this

slc

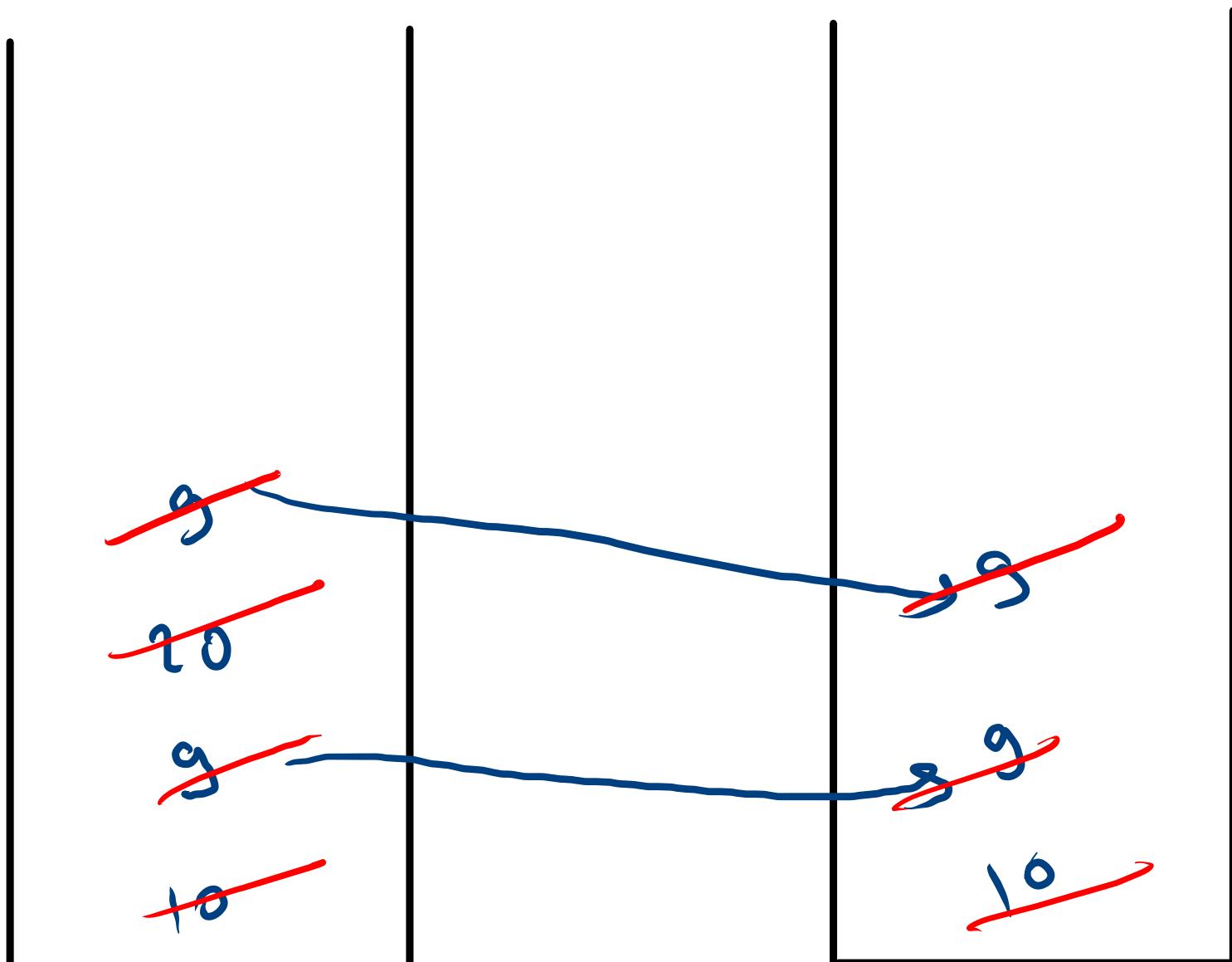
6lc





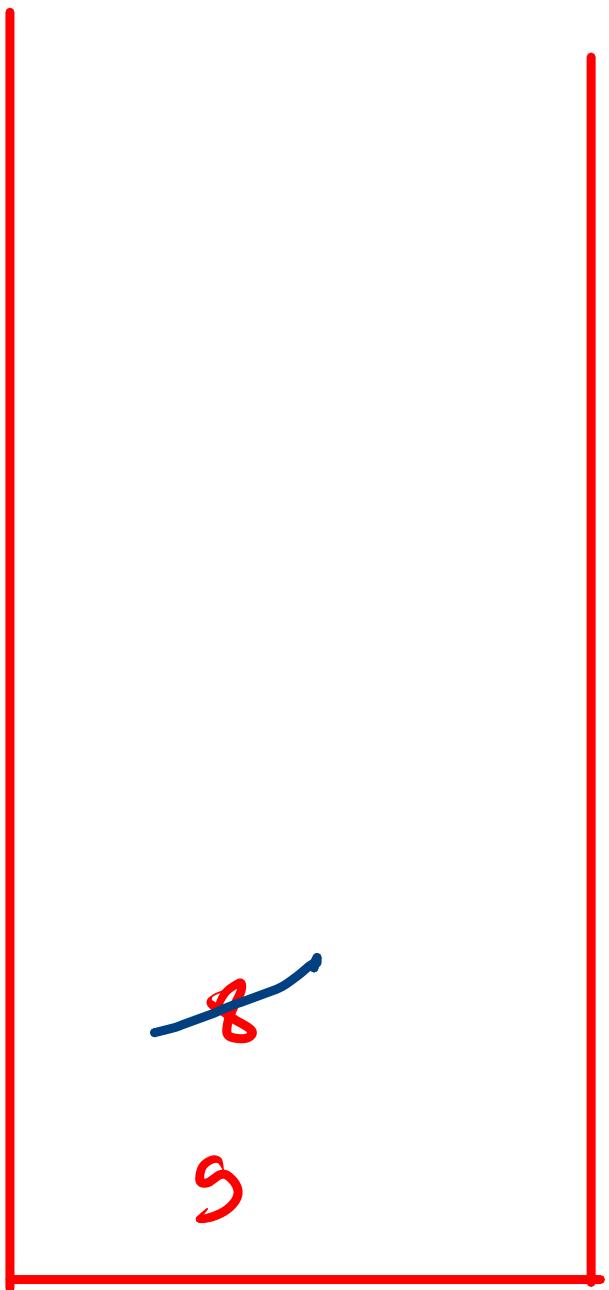
push 10
push 20
top → 20
size 1
min → 1
push 5
min → 4

sum
avg
size



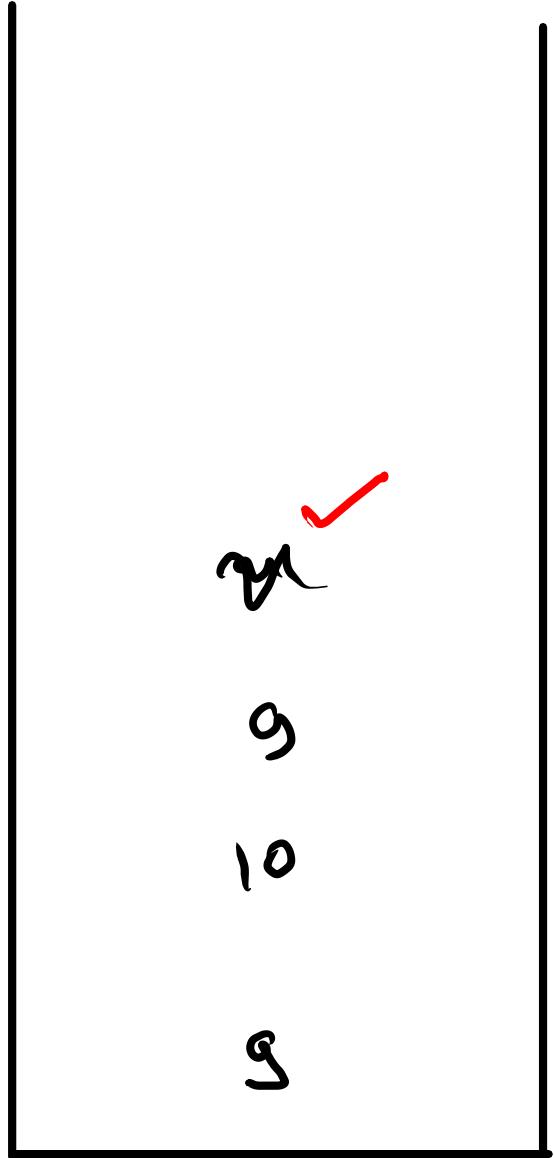
all data

mix data



8
9
3





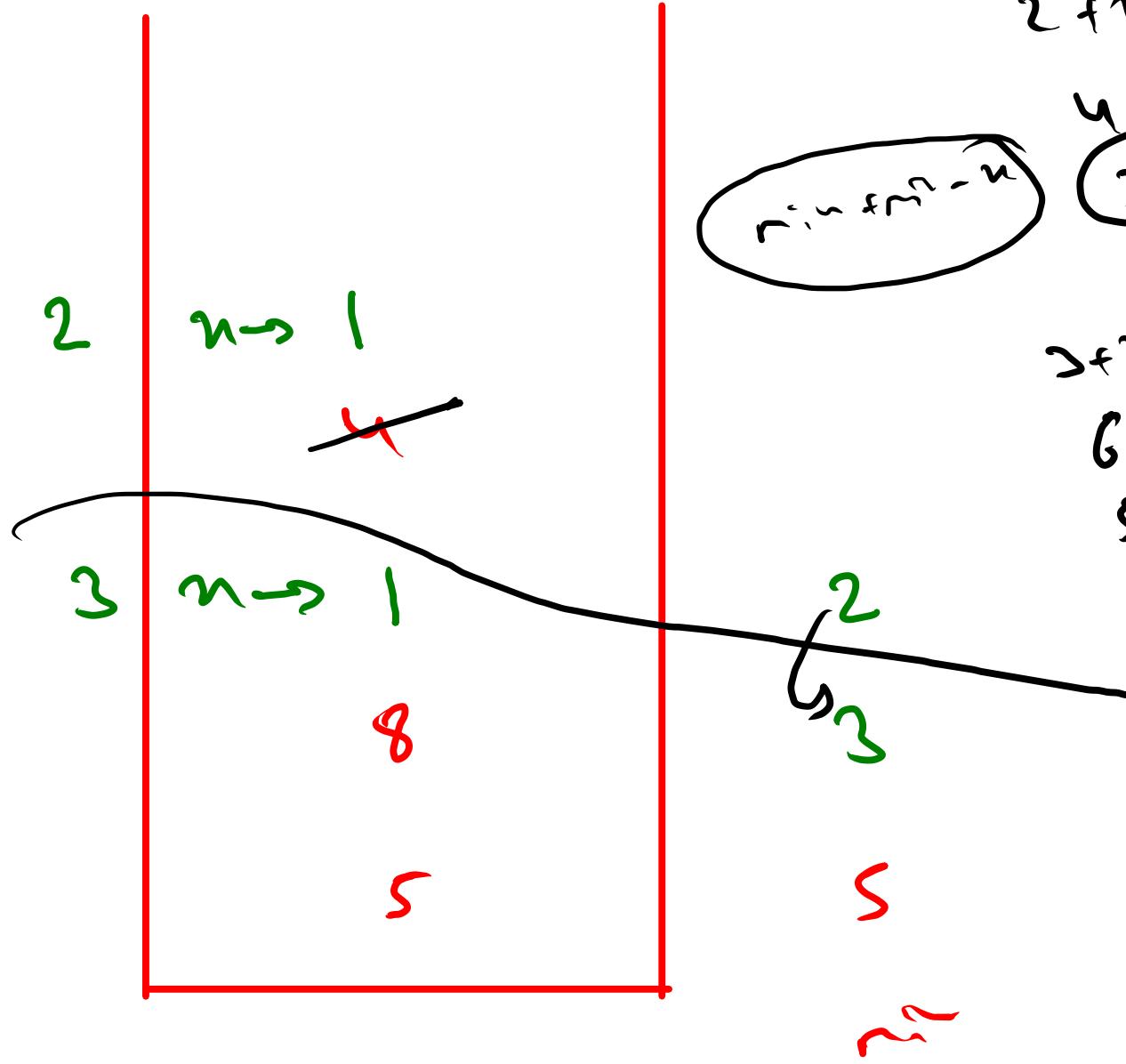
$\text{data} \geq \text{min}$

$\text{data} < \text{min}$

$\text{min} > \text{data}$

$\text{min} - \text{data}$

$$n = \text{data} + (\text{data} - \text{min})$$



$$\min_{\tilde{n}} \text{data} - \tilde{n} = 2$$

$$2+2-1$$

$$4-1$$

$$\min_{\tilde{n}} \text{data} - \tilde{n}$$

$$2+2-1$$

$$6-1$$

$$5$$

$$2$$

$$3$$

$$5$$

$$\tilde{n}$$

$$\text{data} - 3 < \tilde{n}$$

$$\checkmark$$

$$\checkmark n = \text{data} + (\tilde{n} - \min)$$

$$3 + (3-5)$$

$$3 + -2$$

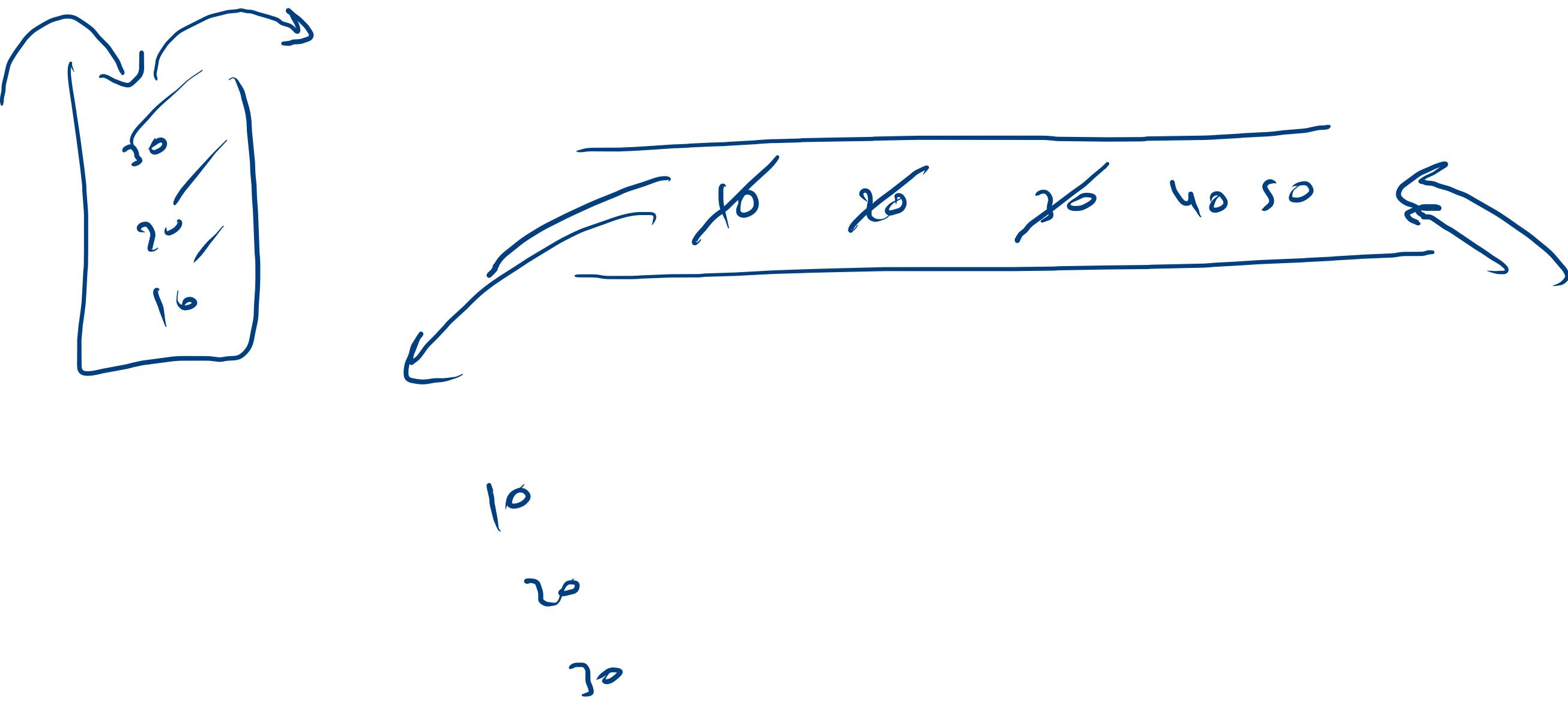
$$1$$

$$\text{data} \geq 2$$

$$n = 2 + (2-2)$$

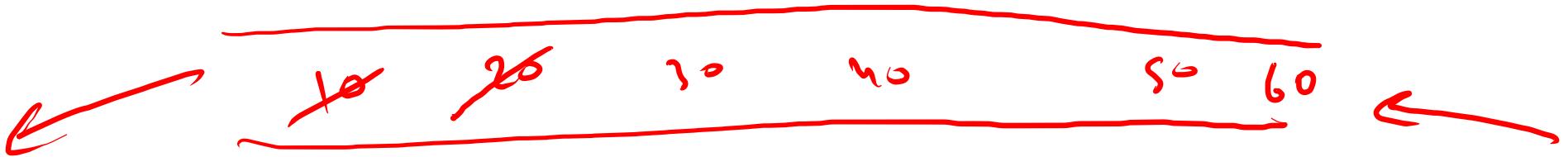
$$2-1$$

$$= 1$$



front

0	1	2	3	4	5	6
60	70	30	40	50		



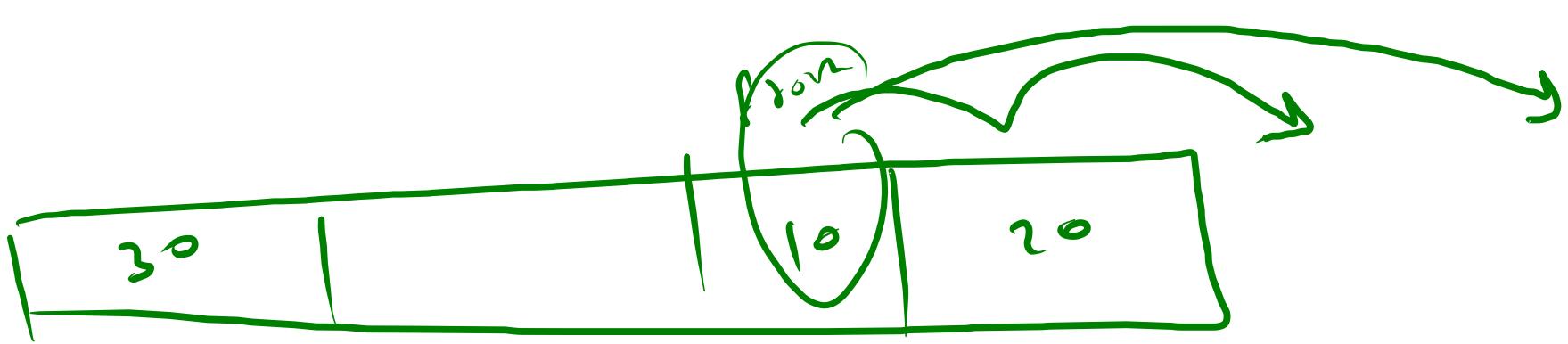
$$size = 4 \times 2^8 \times 8 \times 8 \times 8 \times 8 \times 2^2$$

front + size

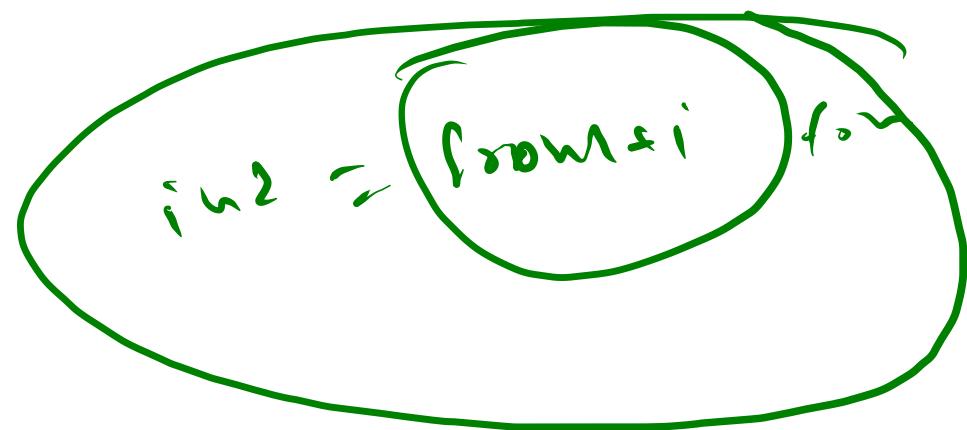
1 - n

20 30 40 10

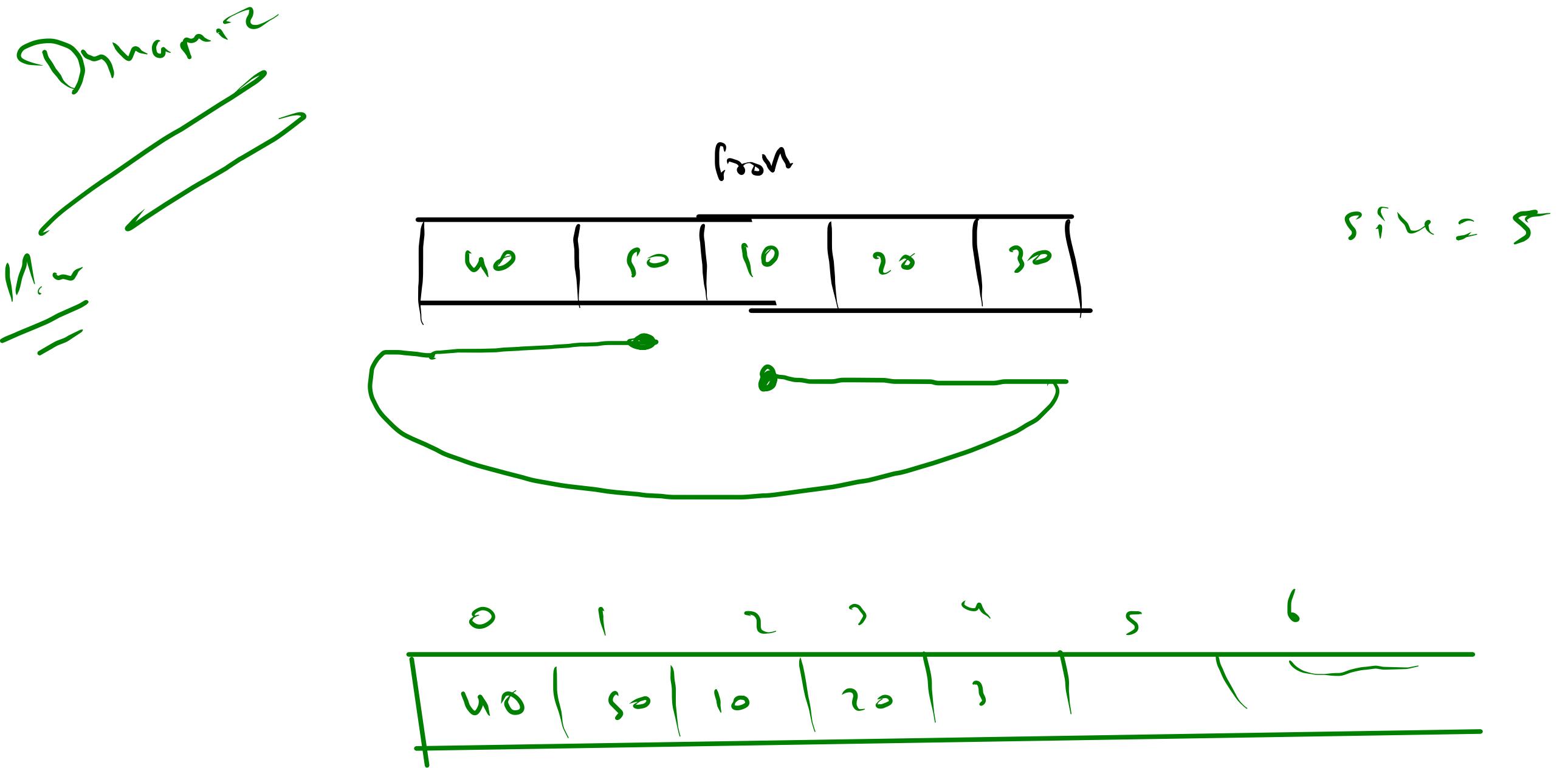
20 30 40 10



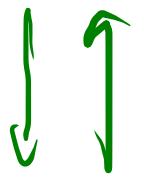
silver



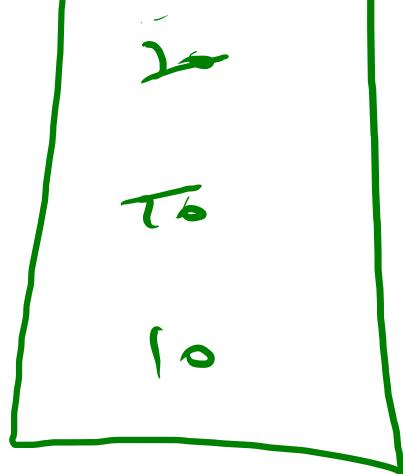
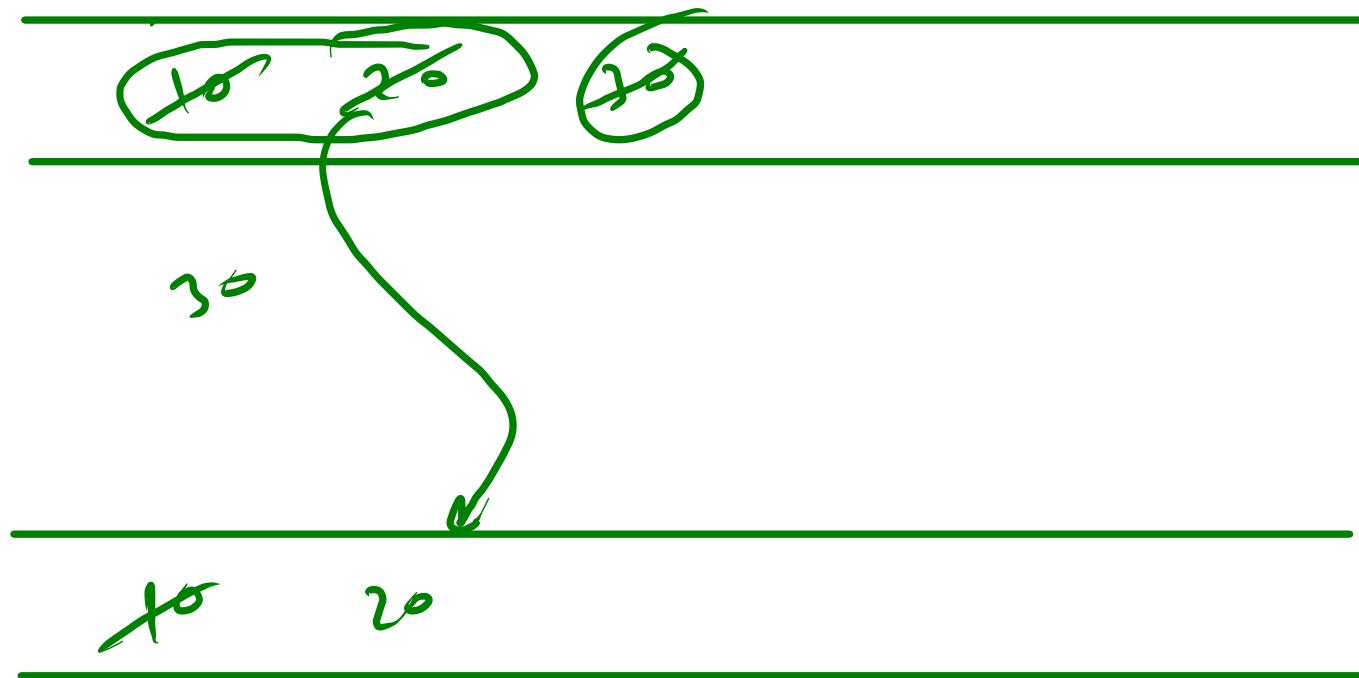
0 1 2
10 26 30



main



helper



~~pop em~~
~~n.s.~~

hd

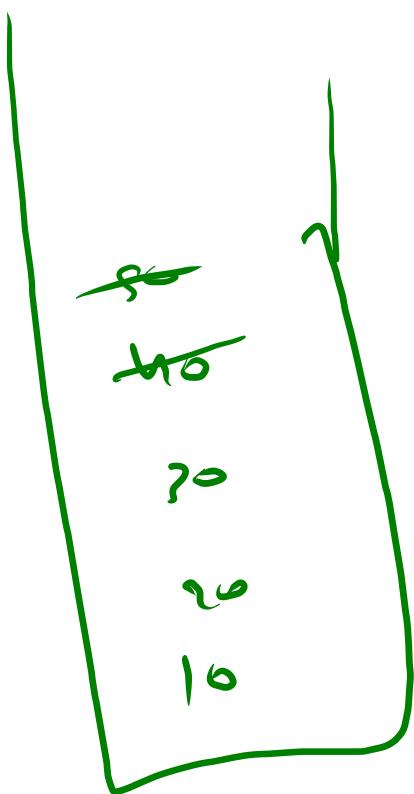


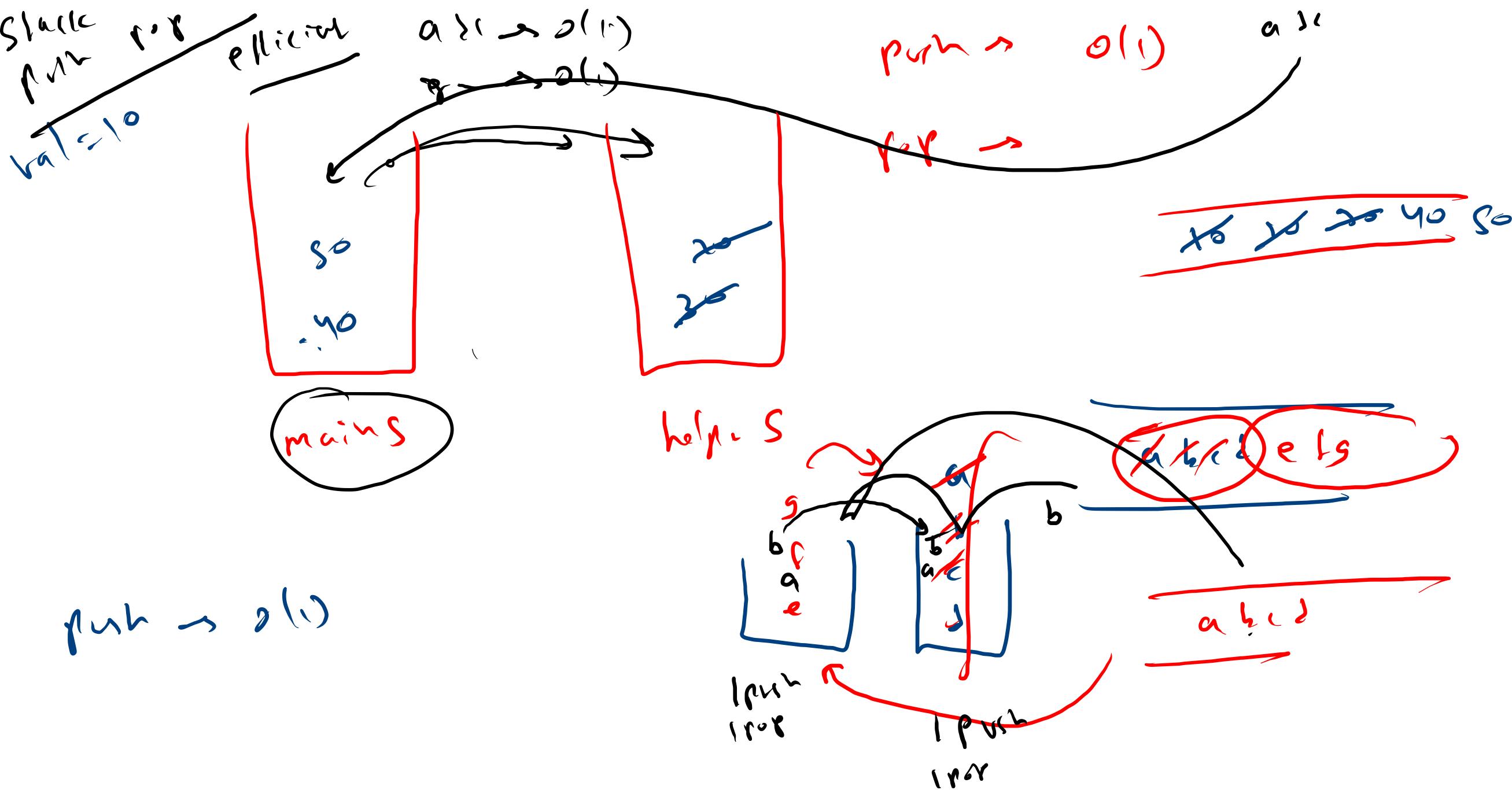
ma



pop $\rightarrow o(1)$

push $\rightarrow o(n)$





1 ~ 2 ~
1 ~ 2 ~

1 ~ 2 ~

1 ~ 2 ~
1 ~ 2 ~
1 ~ 2 ~

tos1

10	20			40	30
----	----	--	--	----	----

tos2

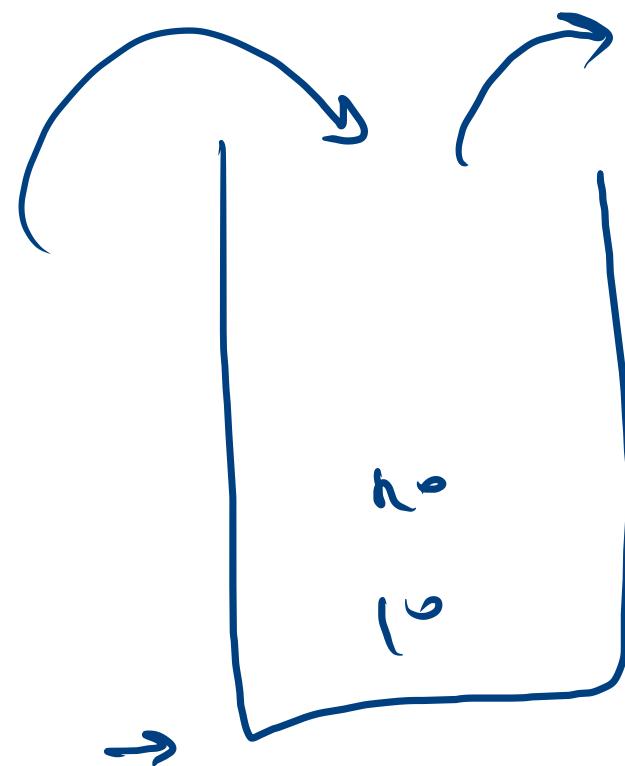
val = 20

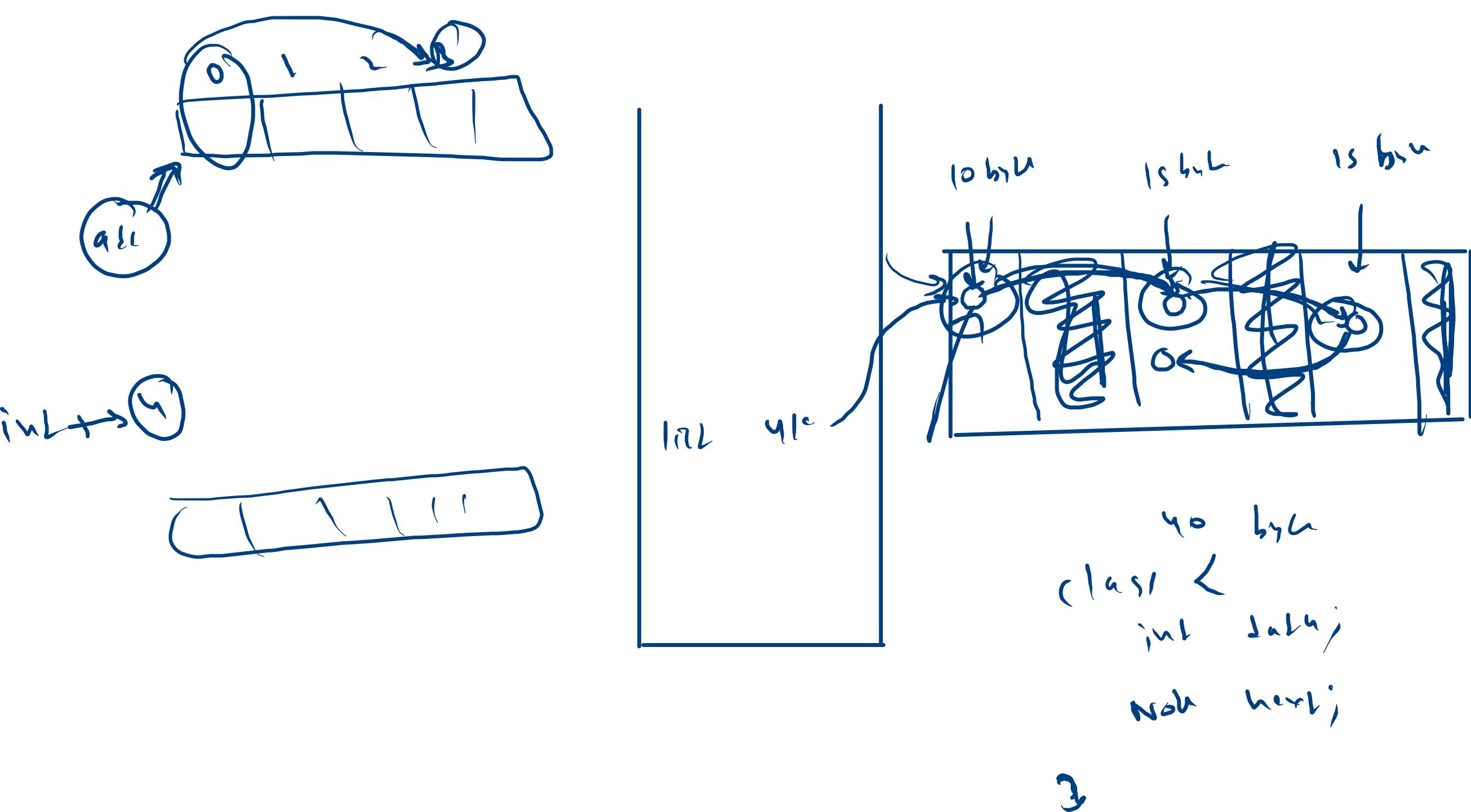
push1(60)

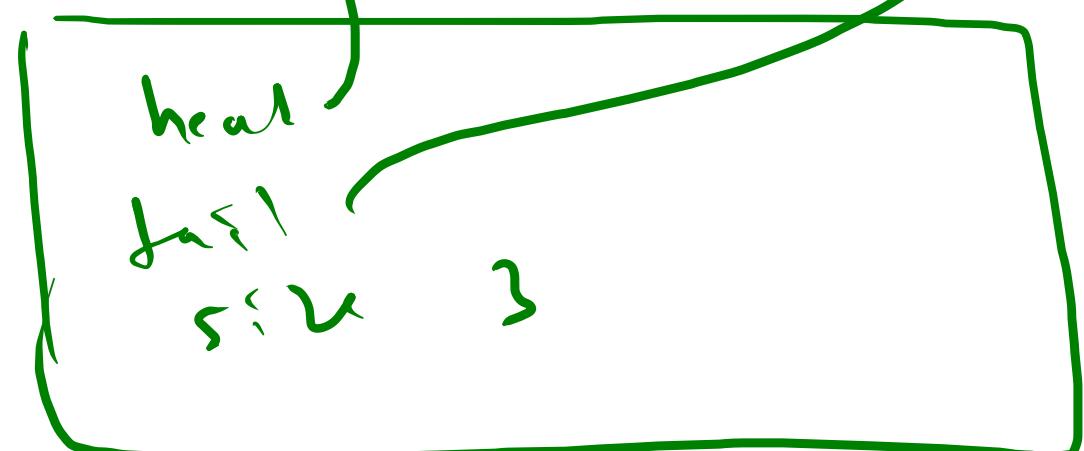
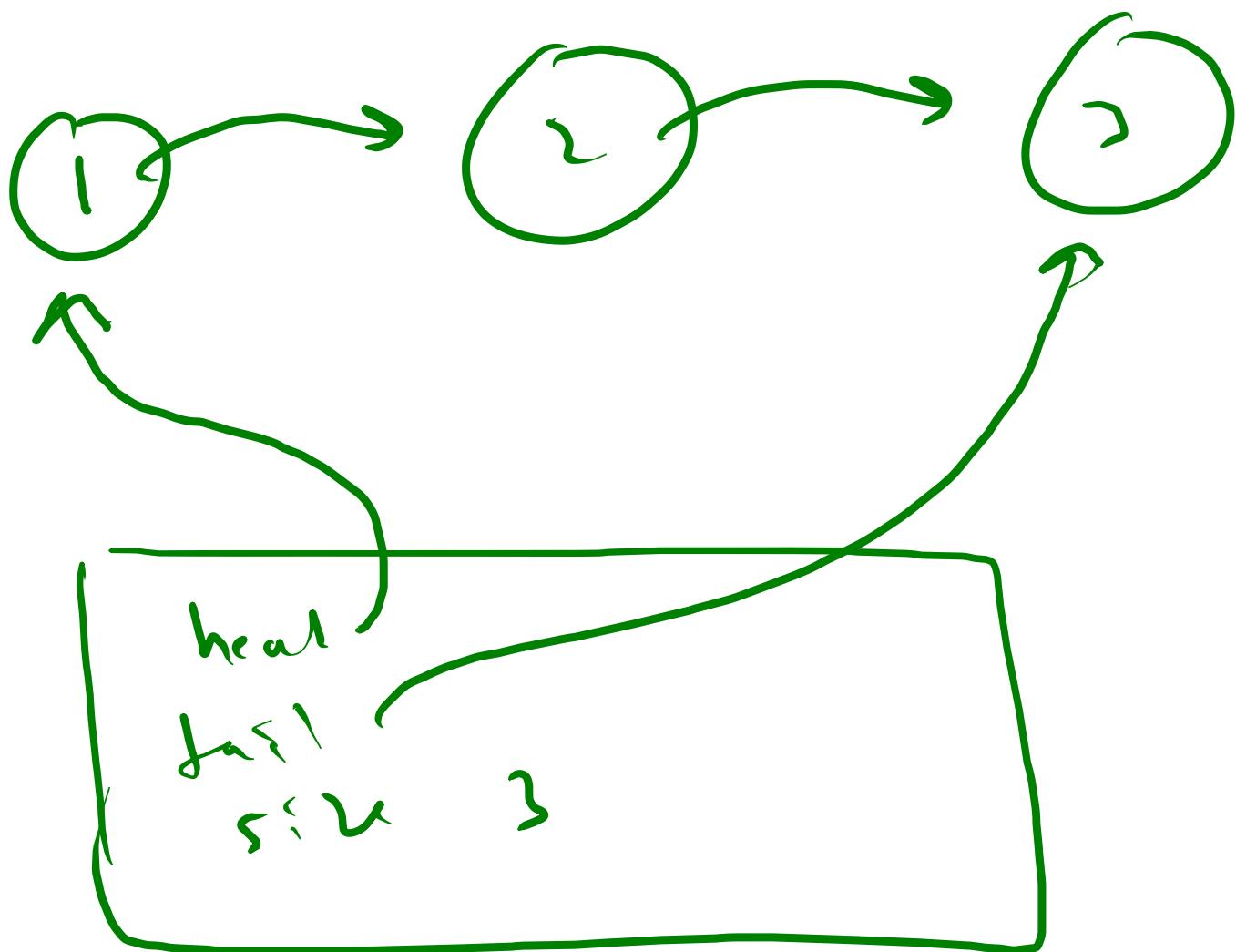
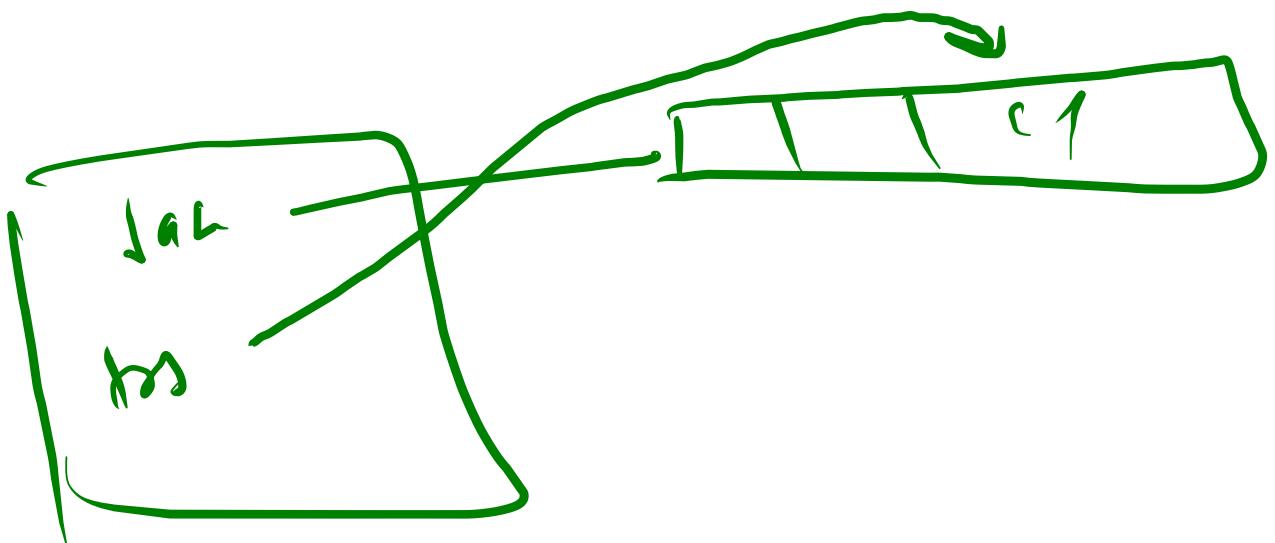
push1(20)

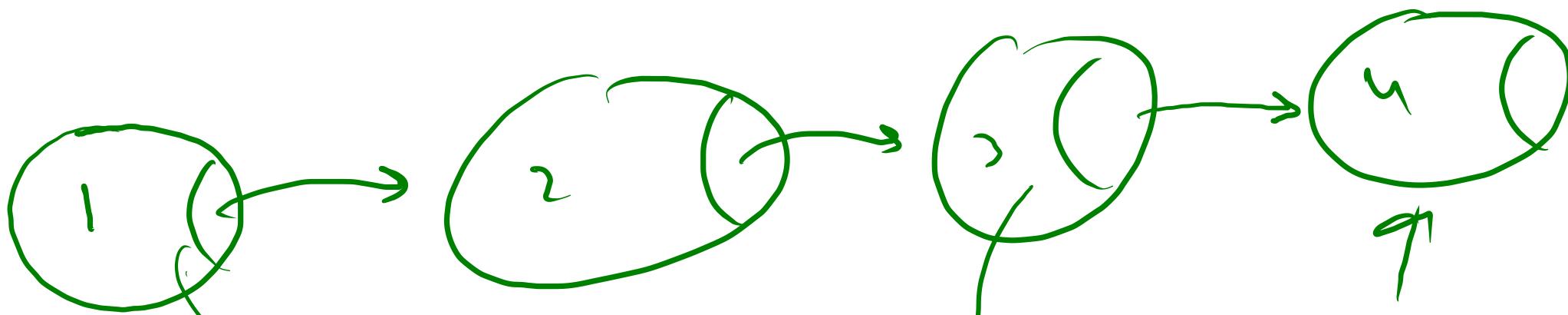
push2(20)

val = 60









value

val = 1



head
tail
size >

```

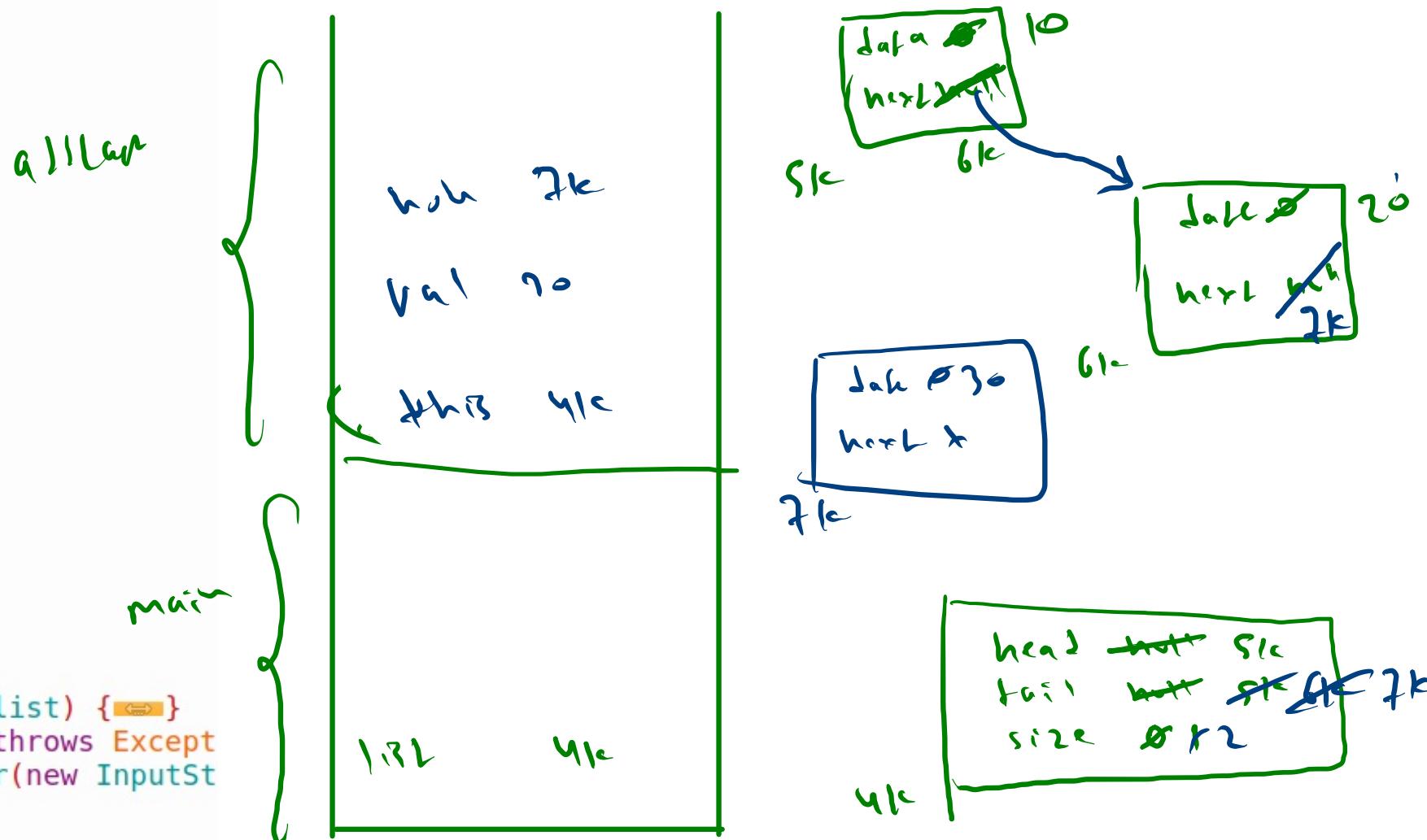
public static class LinkedList {
    Node head;
    Node tail;
    int size;

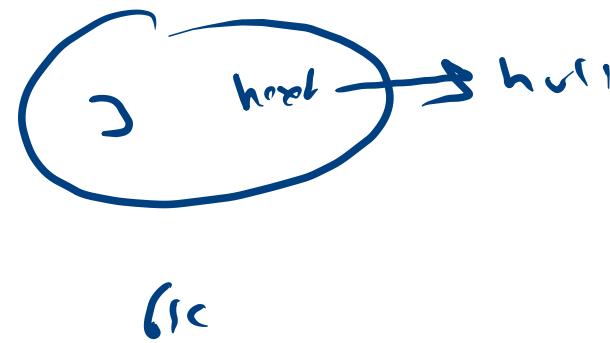
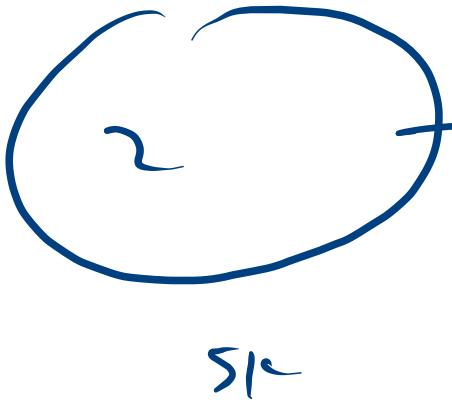
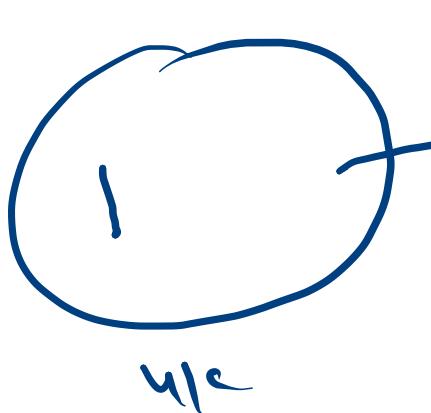
    void addLast(int val) {
        Node node = new Node();
        node.data = val;
        if(size == 0){
            head = tail = node;
            size = 1;
        }else{
            tail.next = node;
            tail = node;
            size++;
        }
    }

    public static void testList(LinkedList list) { }
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        LinkedList list = new LinkedList();

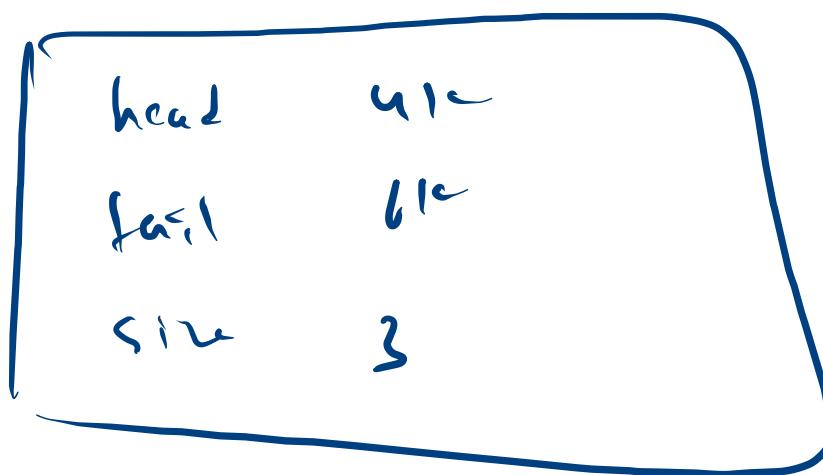
        String str = br.readLine();
        while(str.equals("quit") == false){
            if(str.startsWith("addLast")){
                int val = Integer.parseInt(str.split(" ")[1]);
                list.addLast(val);
            }
            str = br.readLine();
        }
    }
}

```

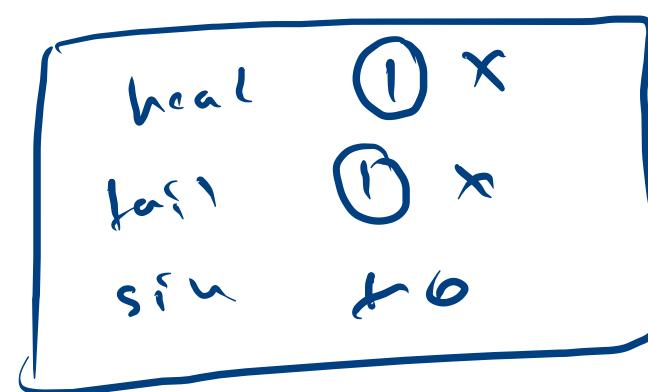
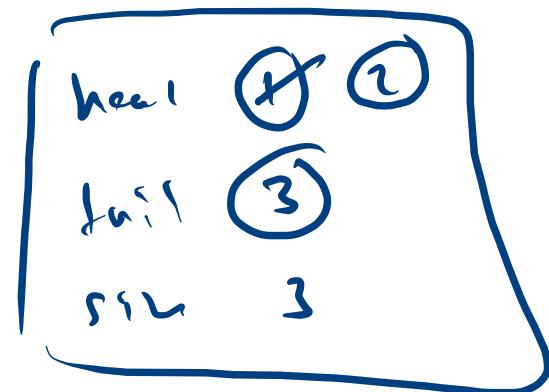
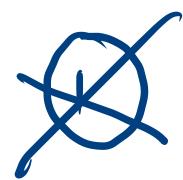
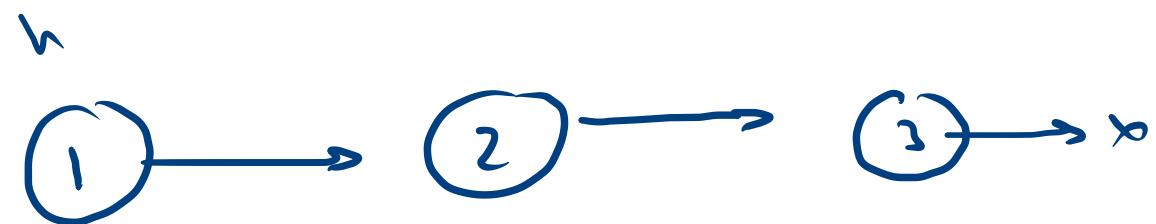


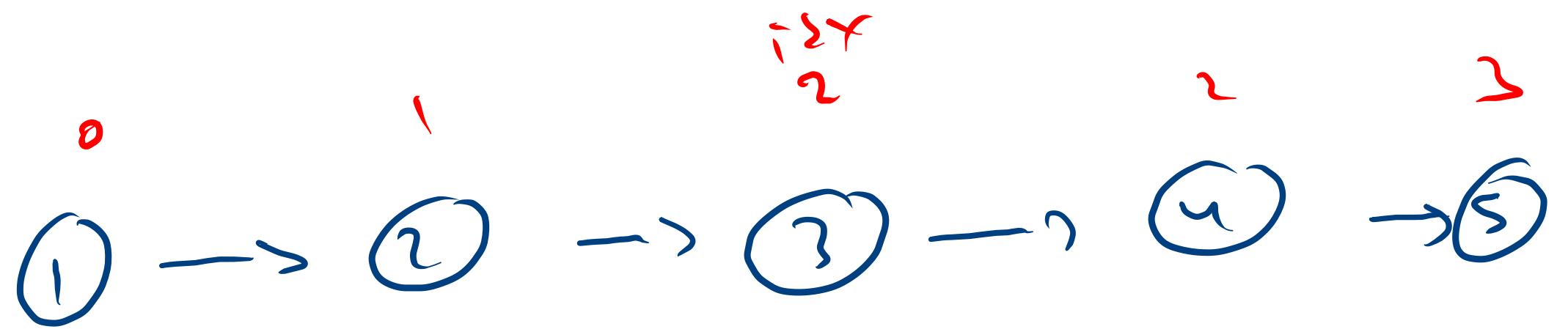


1 2 3



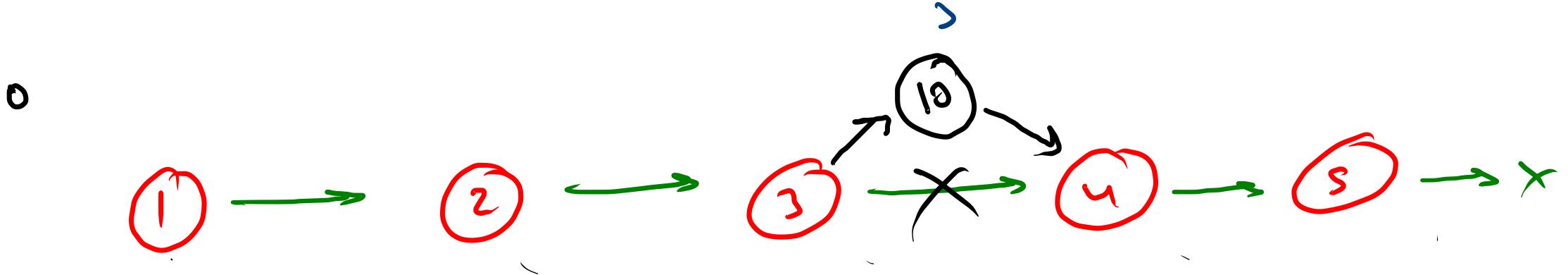
0





tmp

i = 9
2



$i2x = 3$

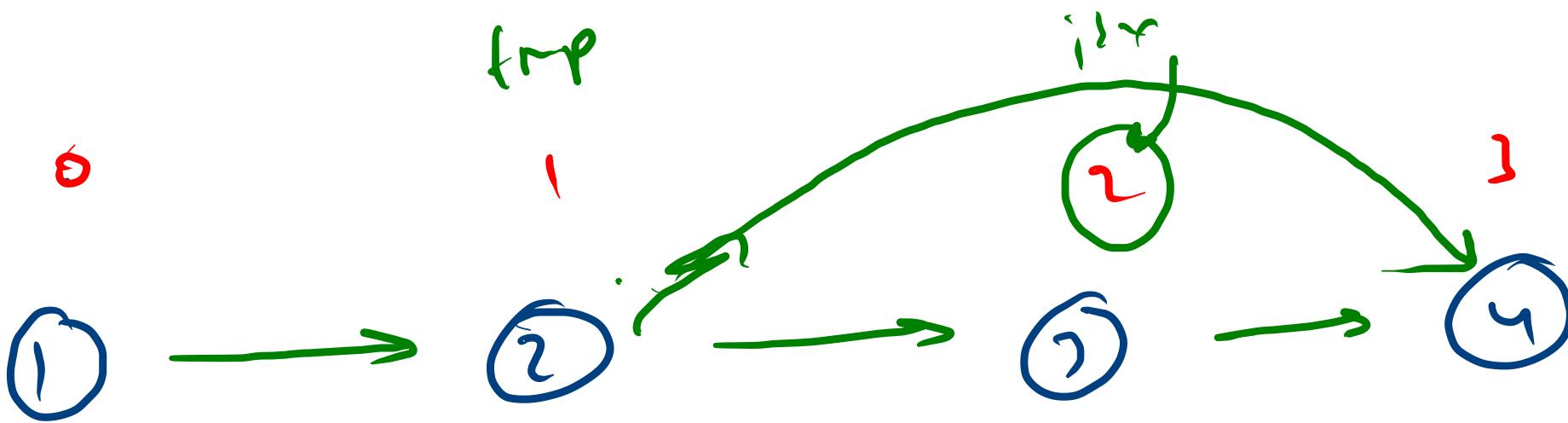
-v

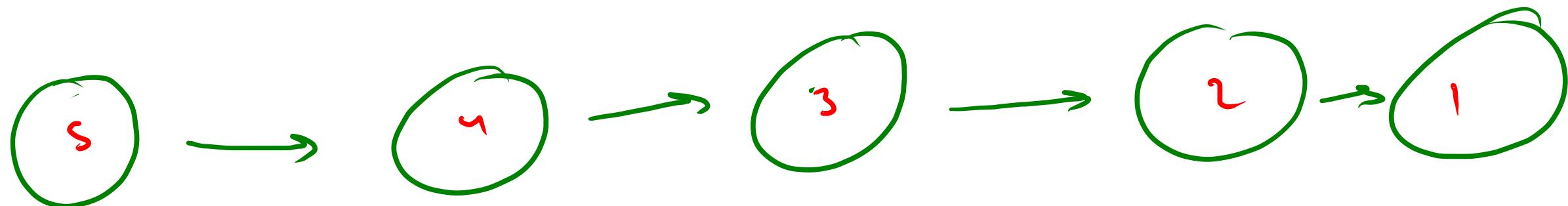
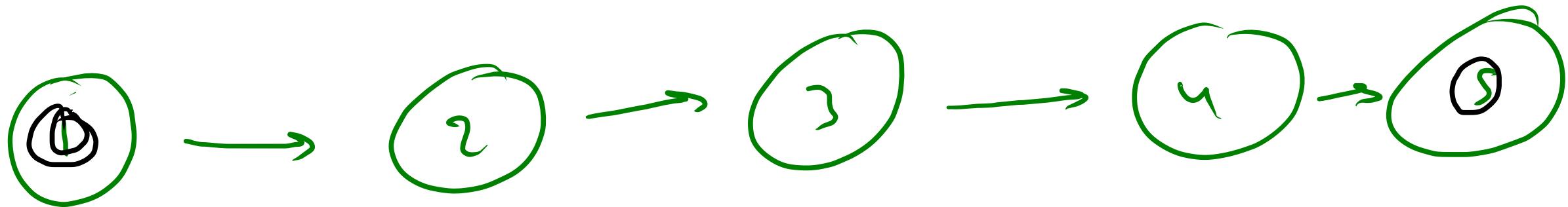
$i2x > size$

0 -- size

$i2x = 0$

$i2x \approx 1$





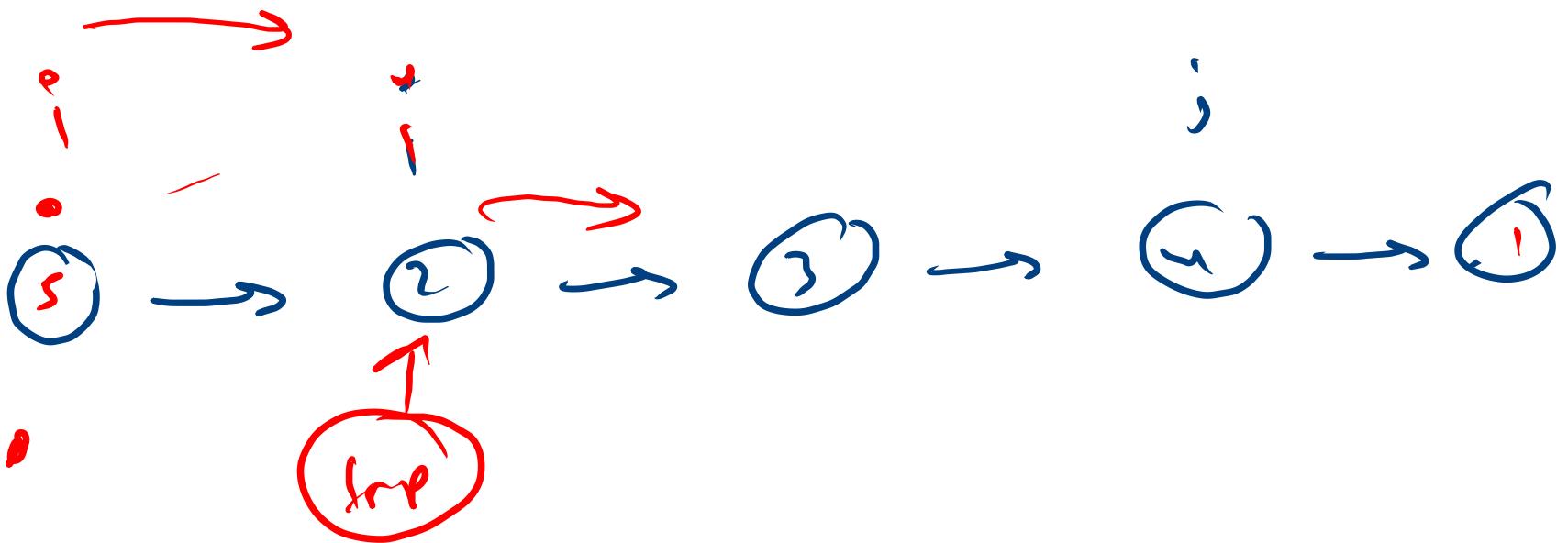
$\text{getAlt}(0) \rightarrow 1$

$\text{getAlt}(n) \rightarrow S$

$\text{getNextAlt}($



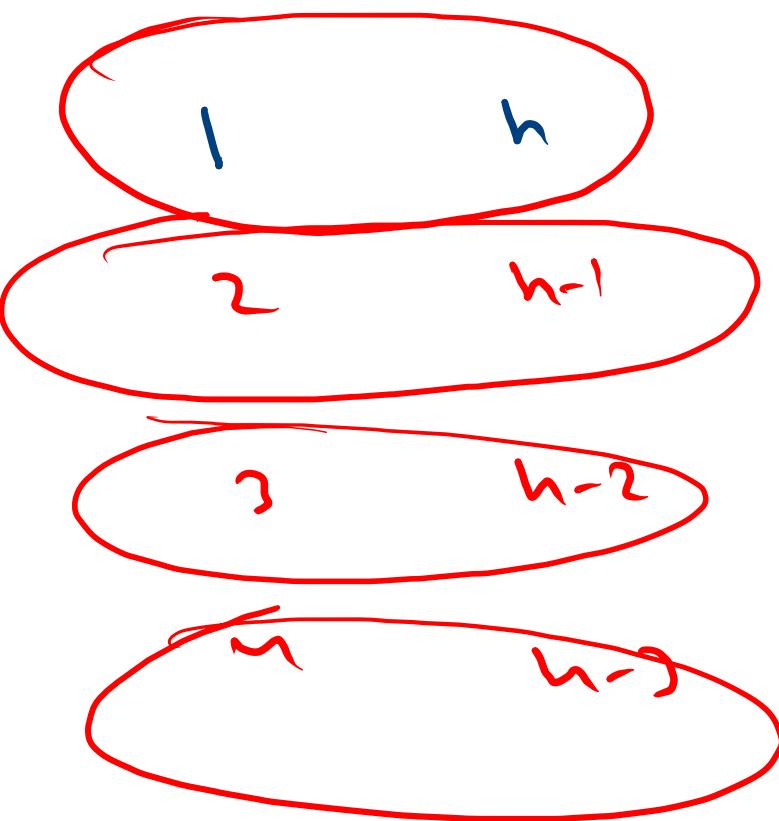
date1
date2

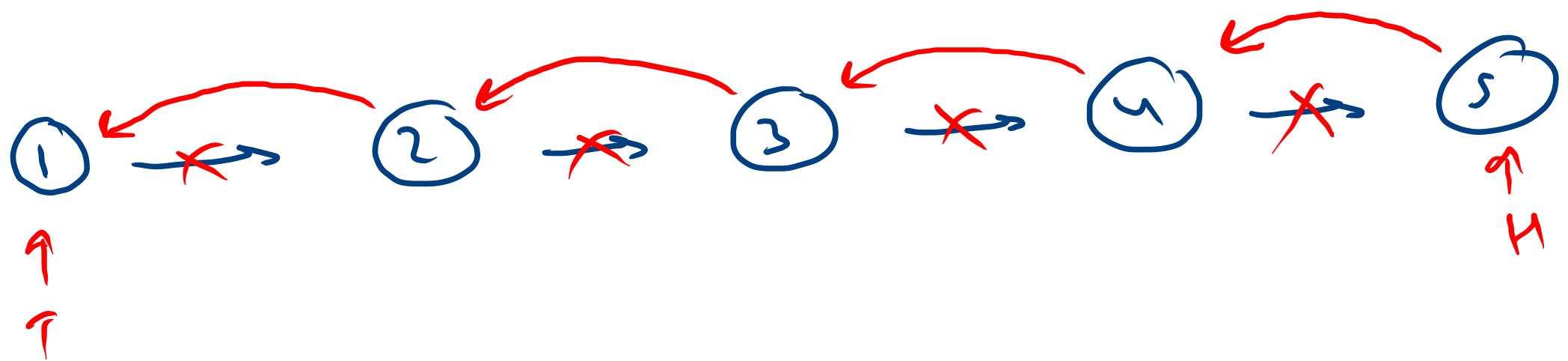
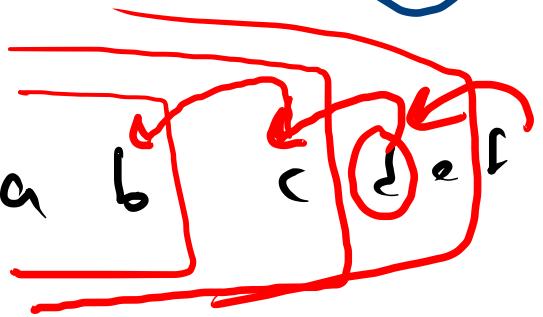
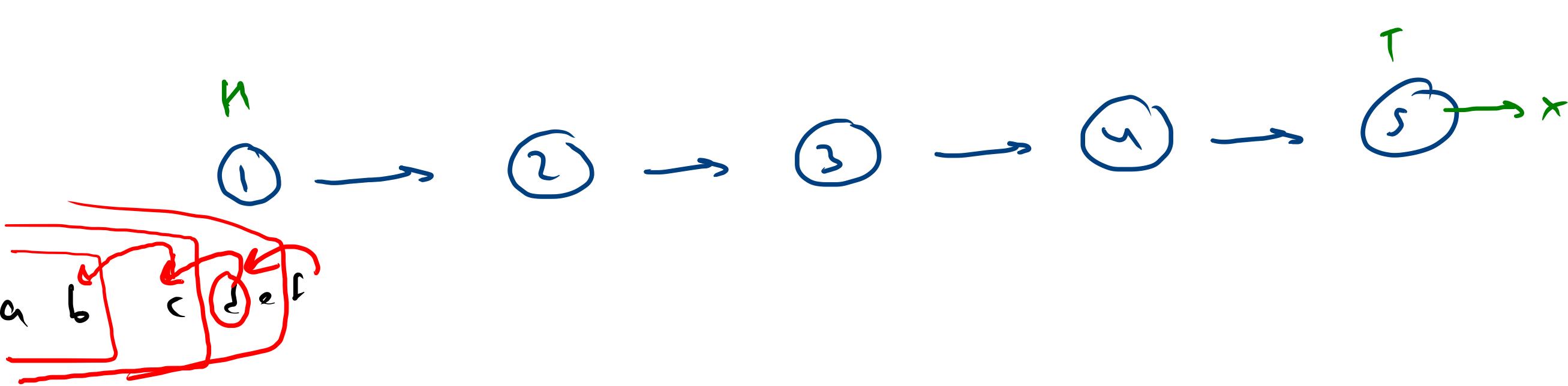


$n_1 \rightarrow n \rightarrow n^2$

n_1

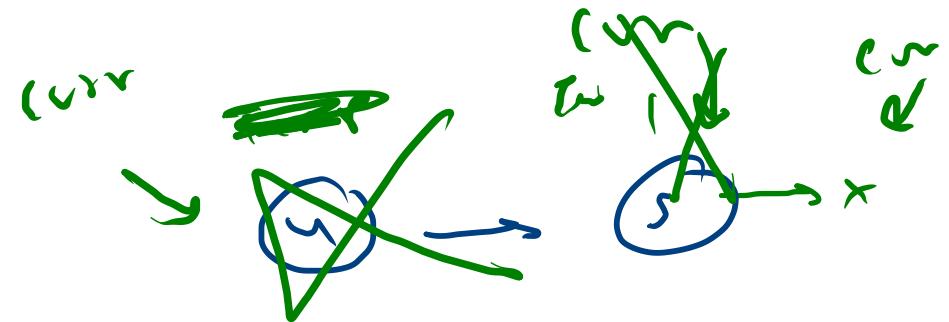
n_2







$\text{prev} \rightarrow \text{null}$

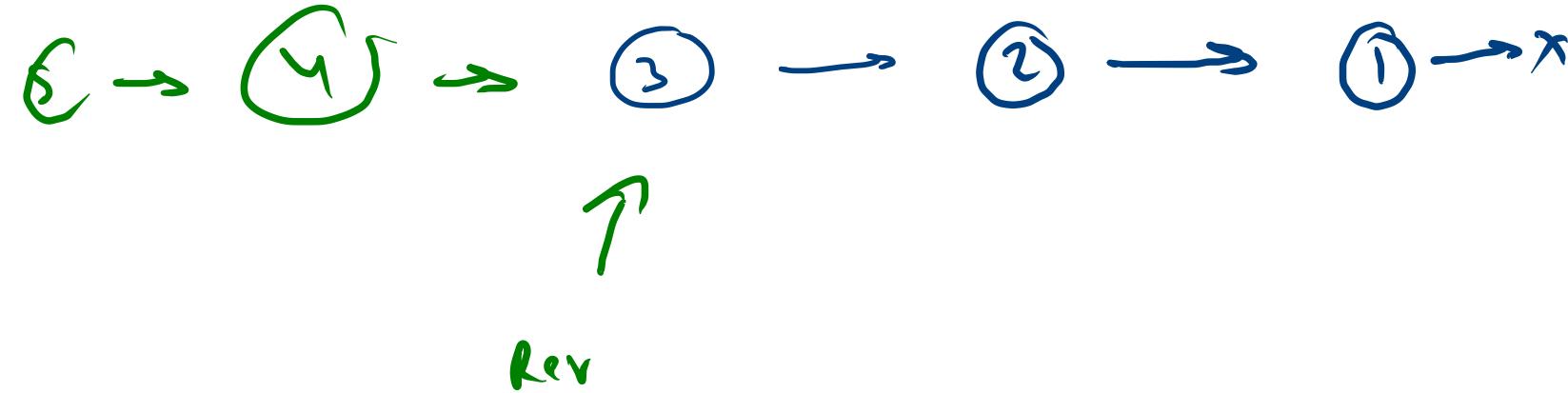


$\text{temp} = \text{curr.next}$

$\text{curr.next} = \text{rev}$

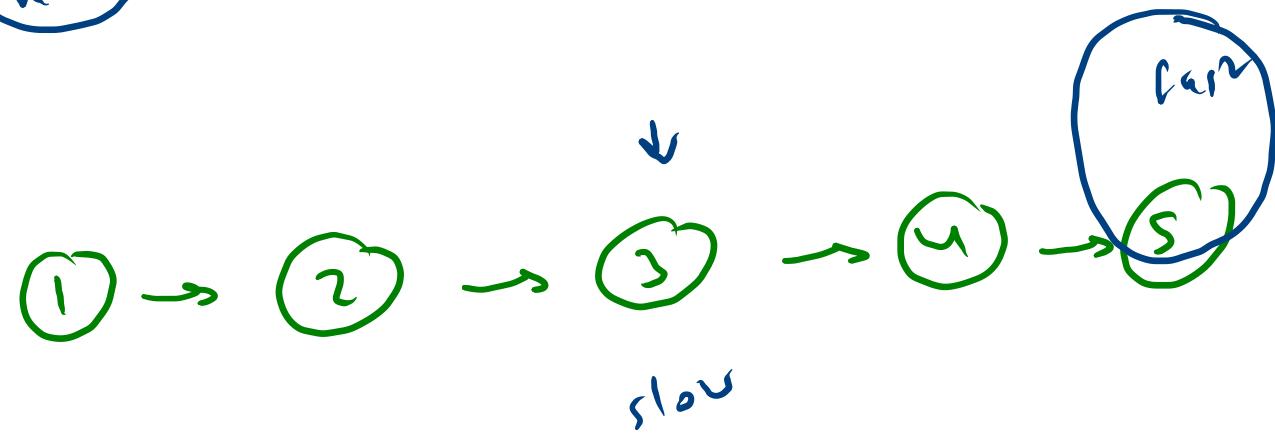
$\text{rev} = \text{curr}$

$\text{curr} = \text{temp}$

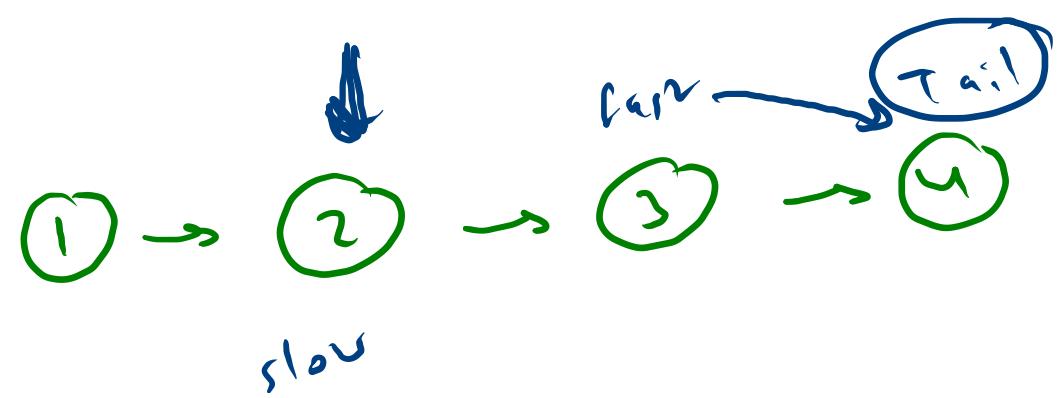


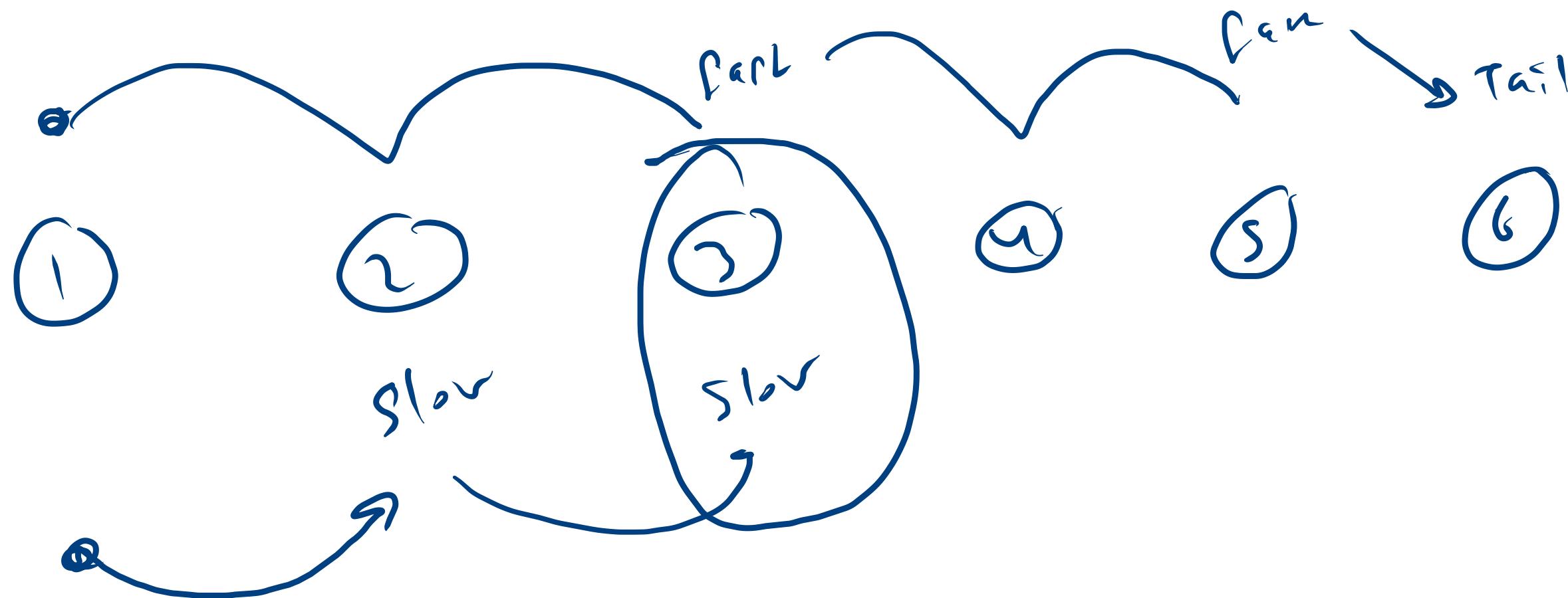
fast-head
head

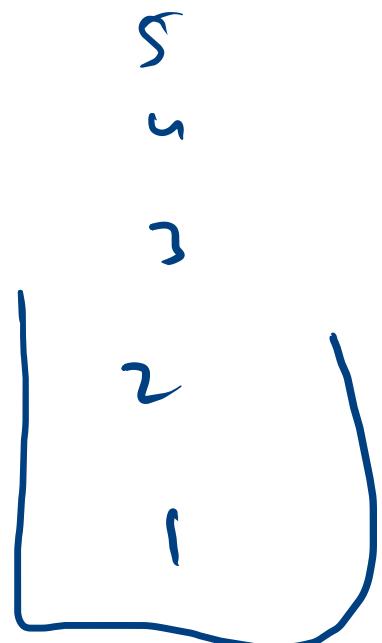
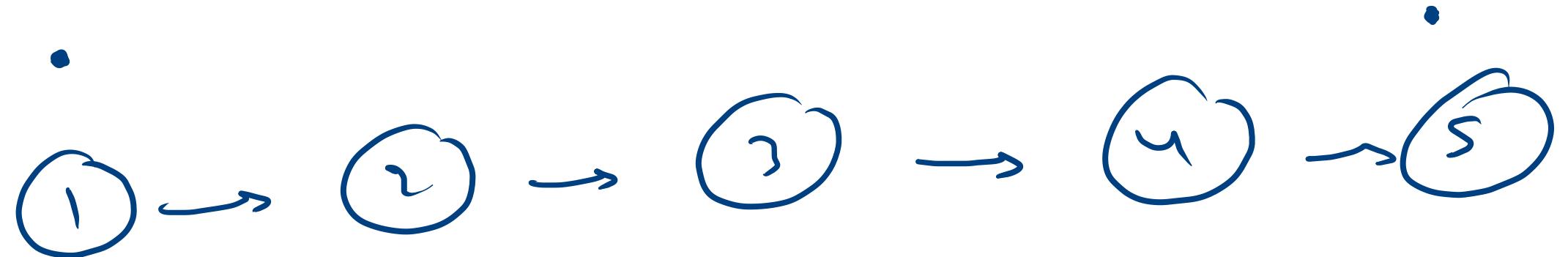
fast-head = tail

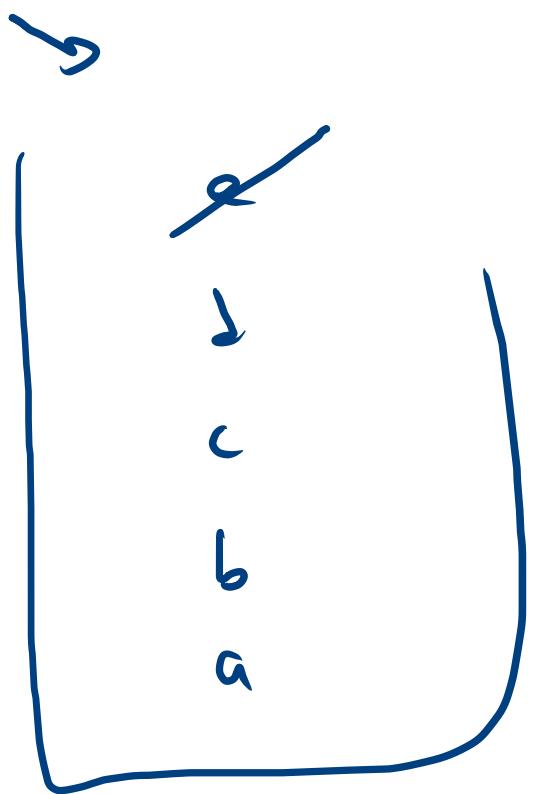
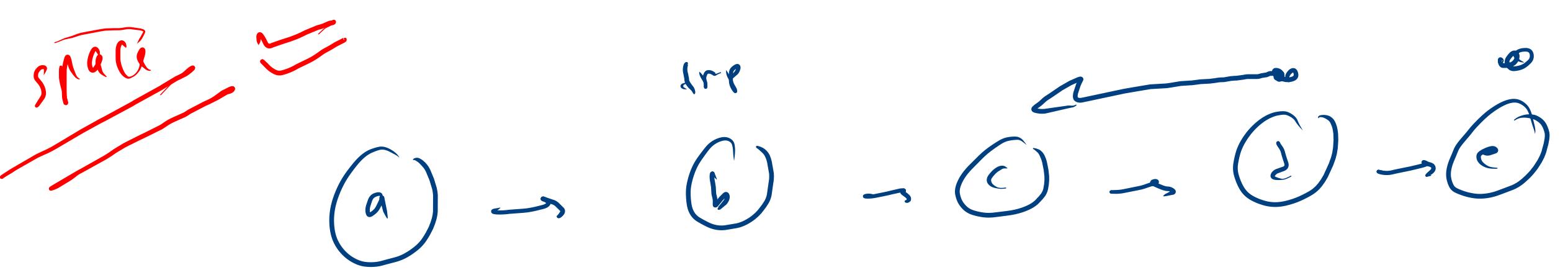


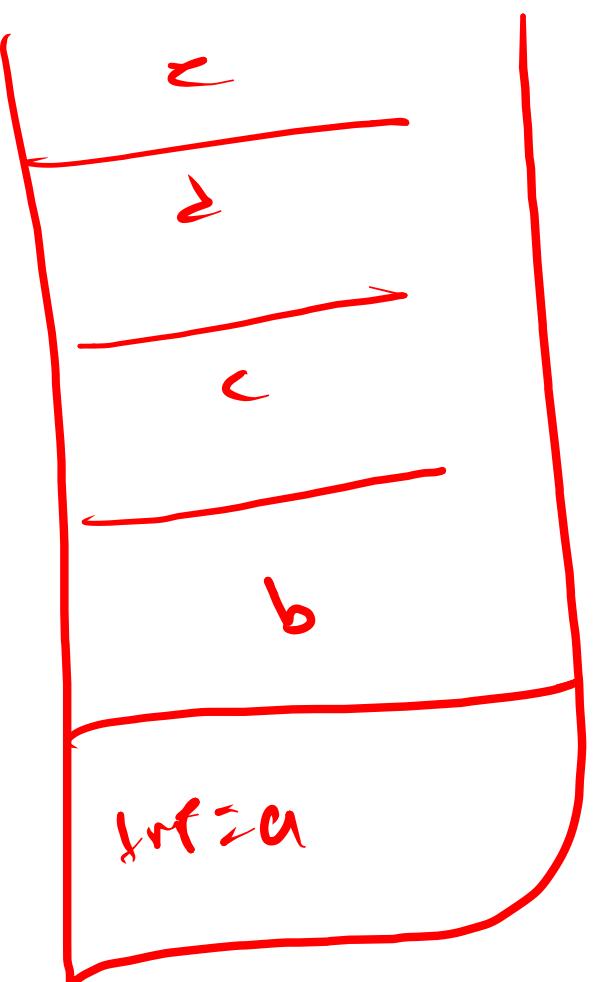
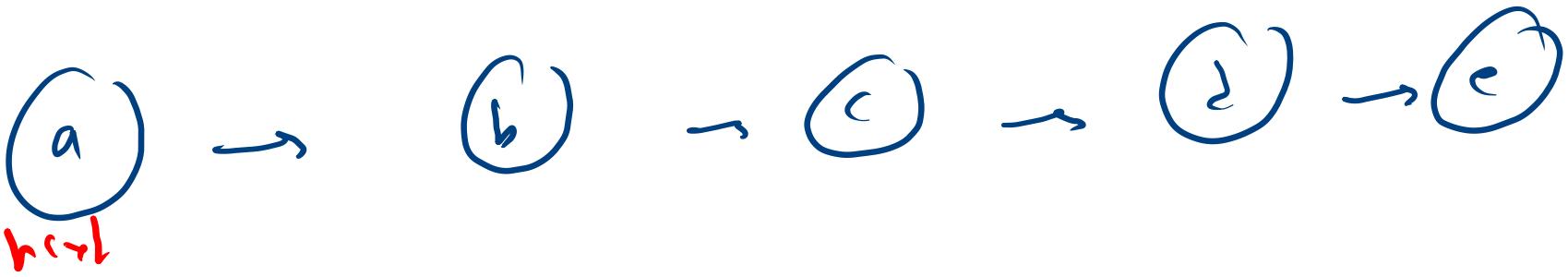
fast-head = tail

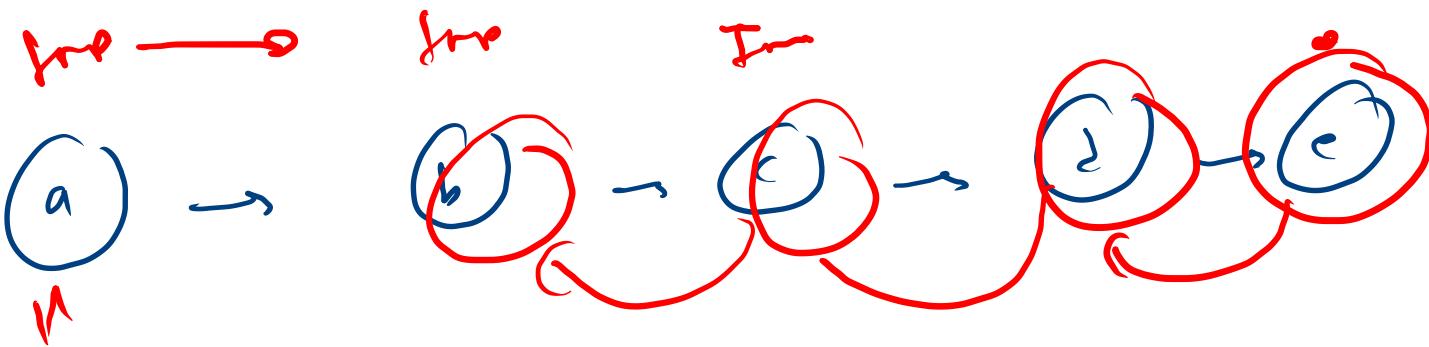










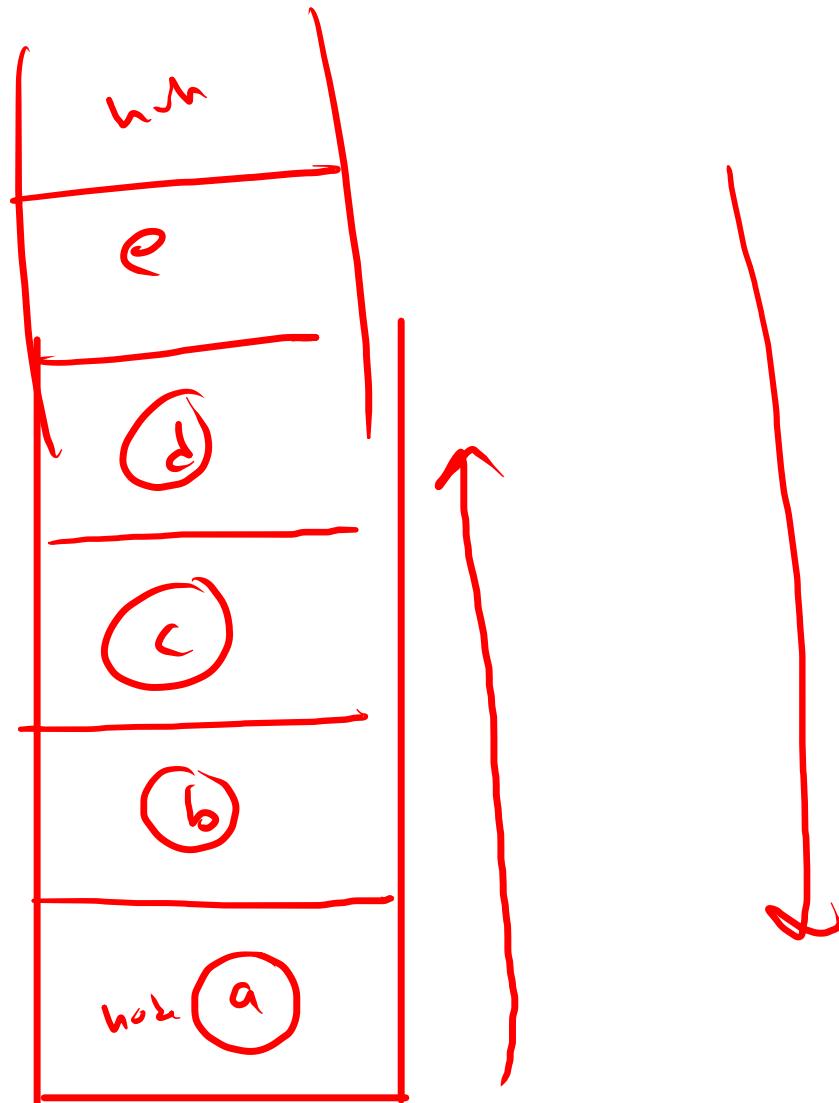


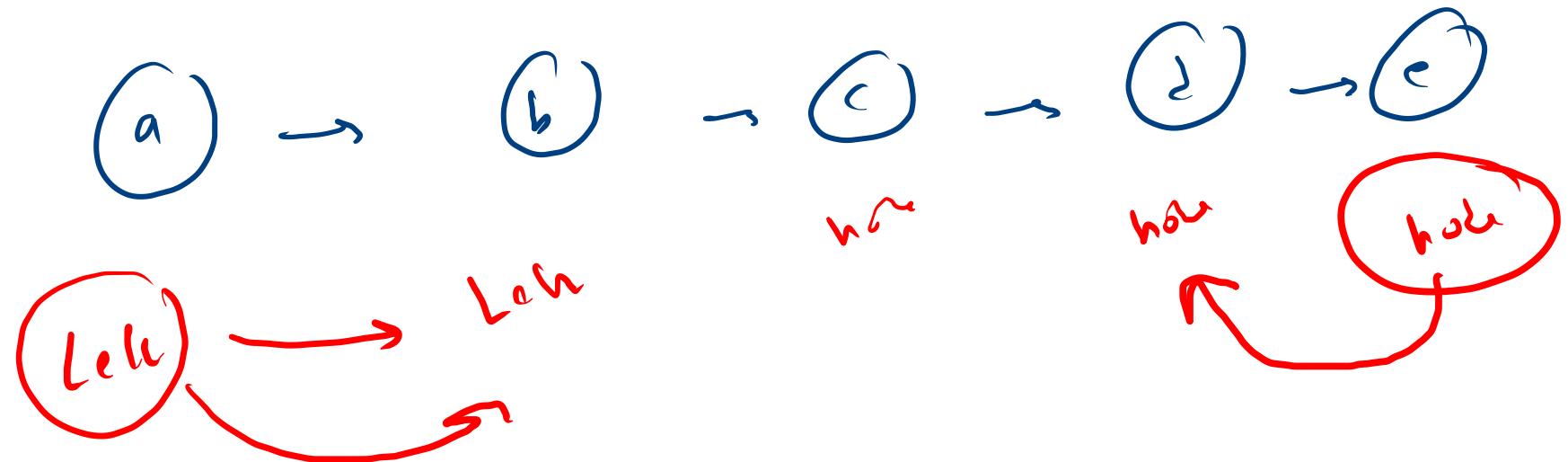
```

public static boolean ispalin(Node node){
    if(node == null){
        return true;
    }
    ispalin(node.next);
}

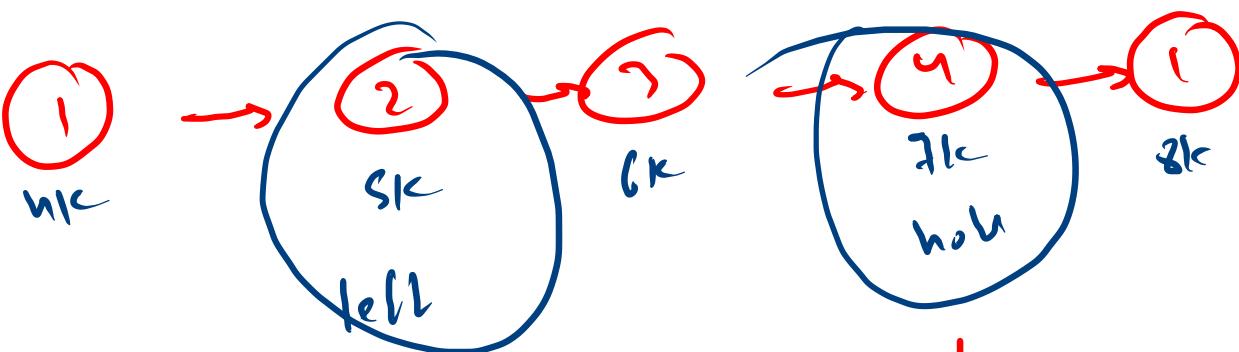
public boolean IsPalindrome() {
    return ispalin(head);
}

```





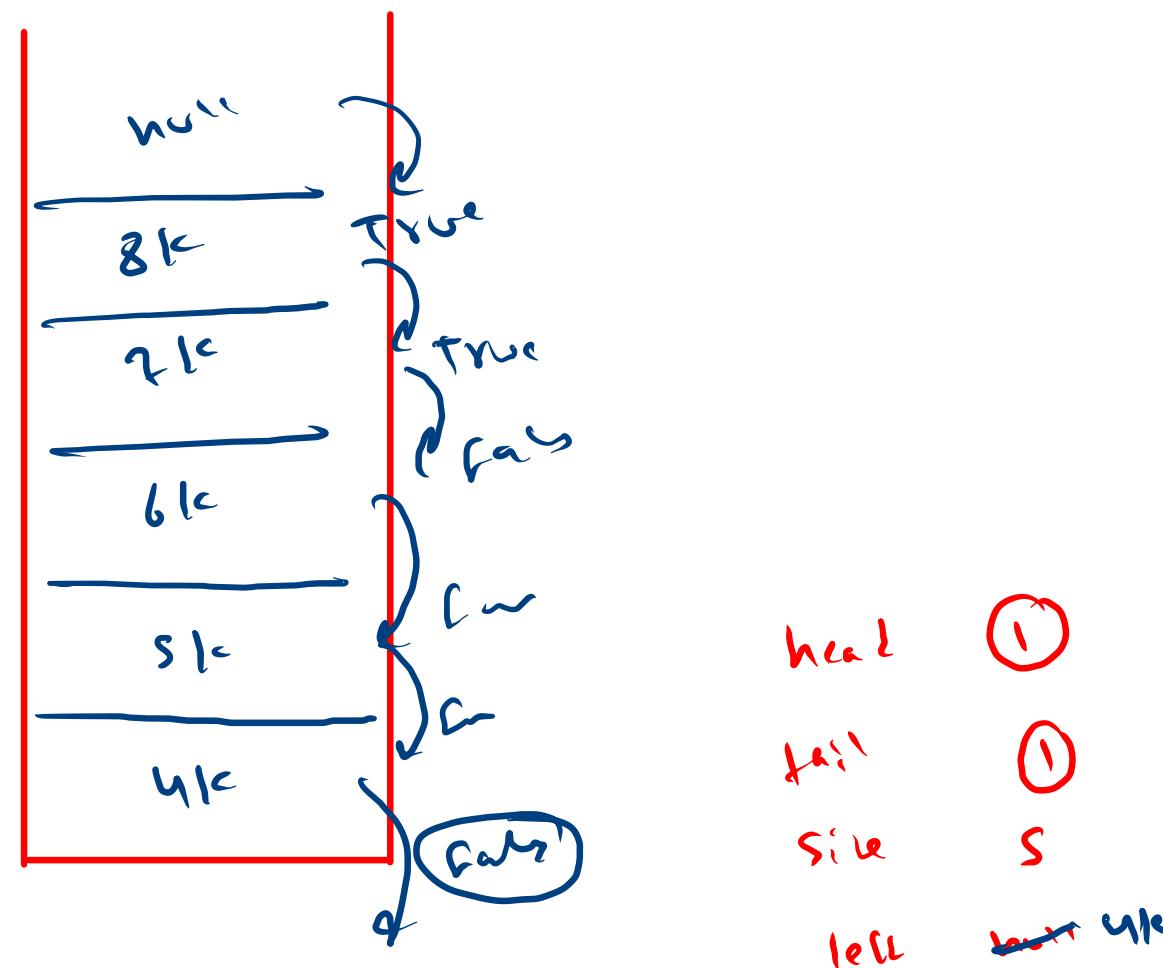
heat
sea
size
left → heat



```

public boolean ispalin(Node node){
    if(node == null){
        left = head;
        return true;
    }
    True/False
    if(ispalin(node.next) == false) return false;
    ✓ if(node.data != left.data) return false;
    left = left.next;
    return true;
}
Node left;
public boolean IsPalindrome() {
    return ispalin(head);
}

```



[2] [3]

target \rightarrow 7

5

6)

$2+2+2 \rightarrow 7$
 $2+2+2 \rightarrow 7$
 $3+2+2 \rightarrow 7$

0	1	2	3	4	5	6	7
1	0	1	1	1	1	2	1

2

3

2+2+2

2+3

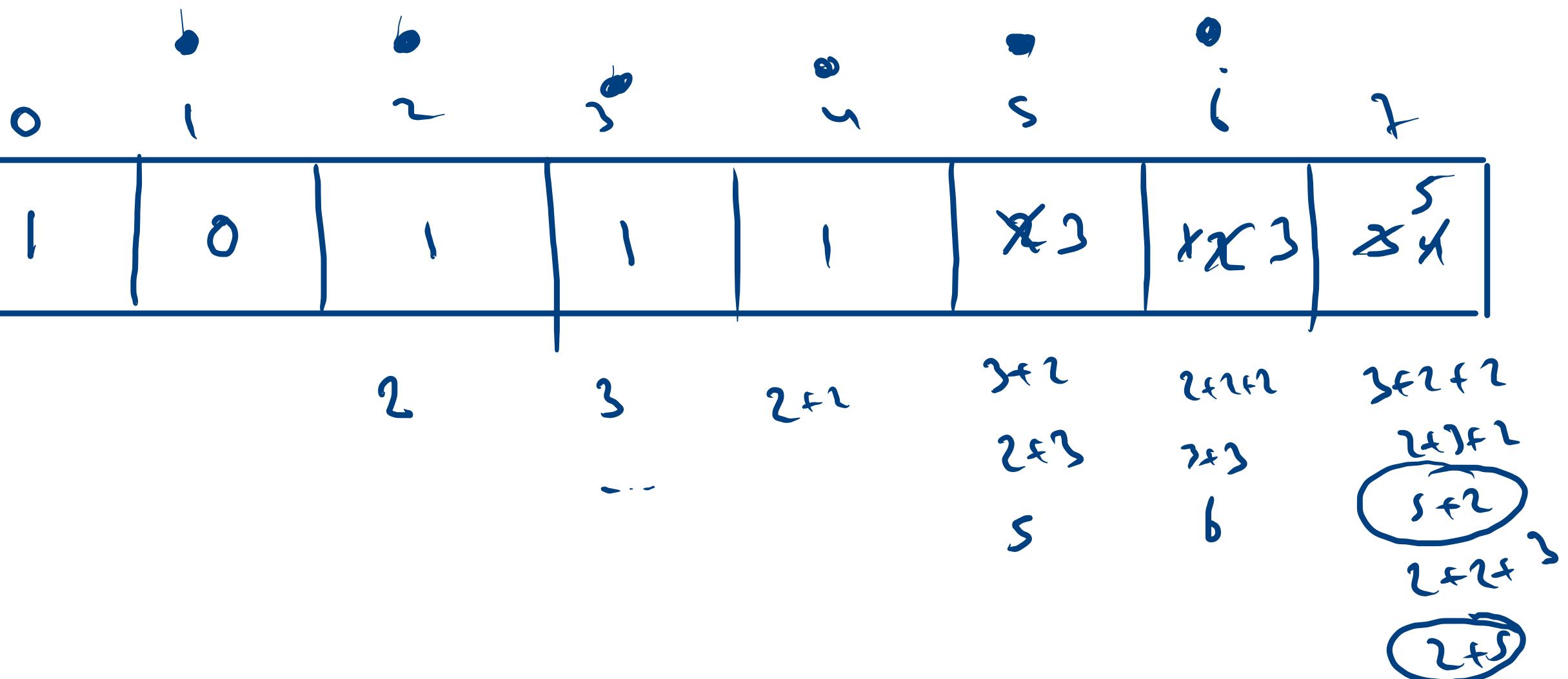
2+1+2

2+2+3

3+3

[2 3 5 6]

target \rightarrow 7



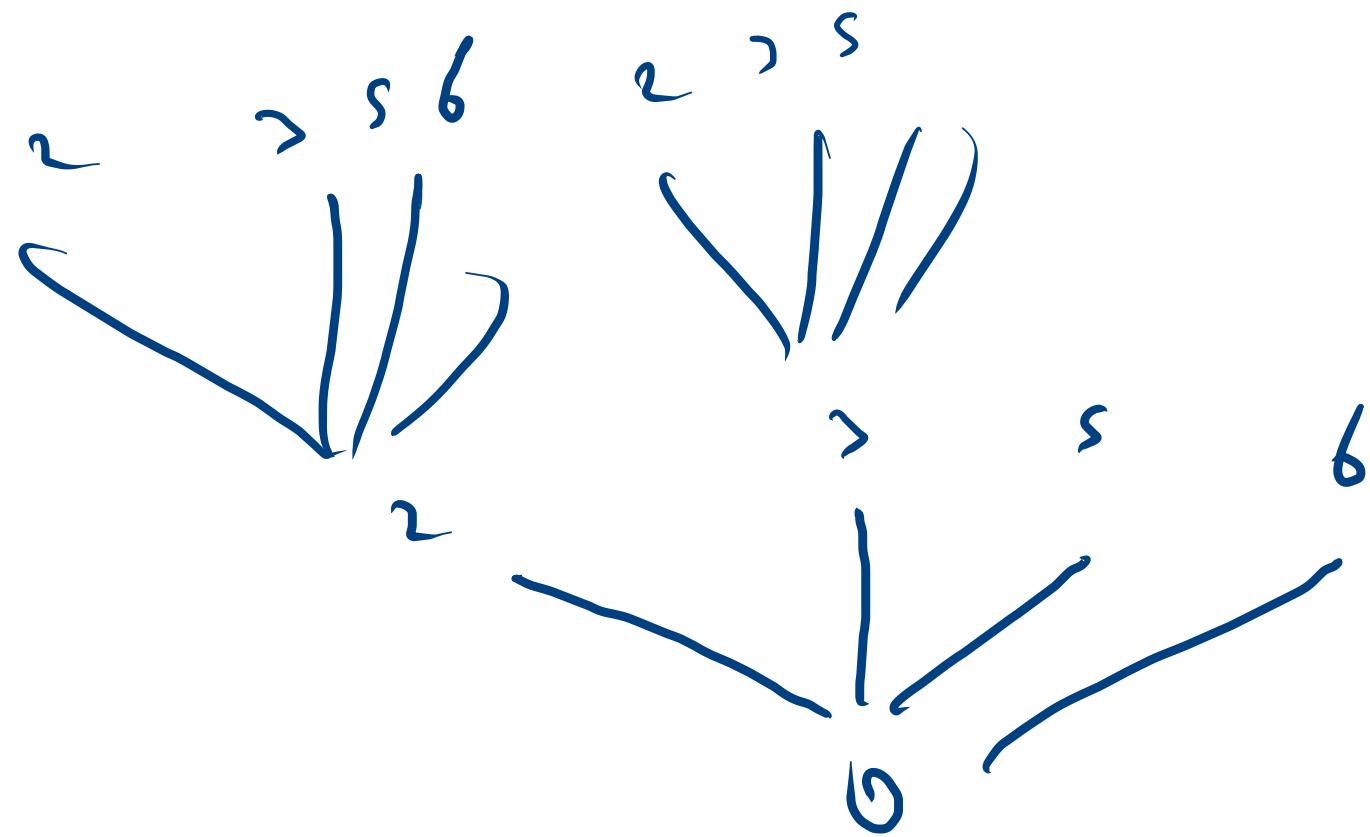
2

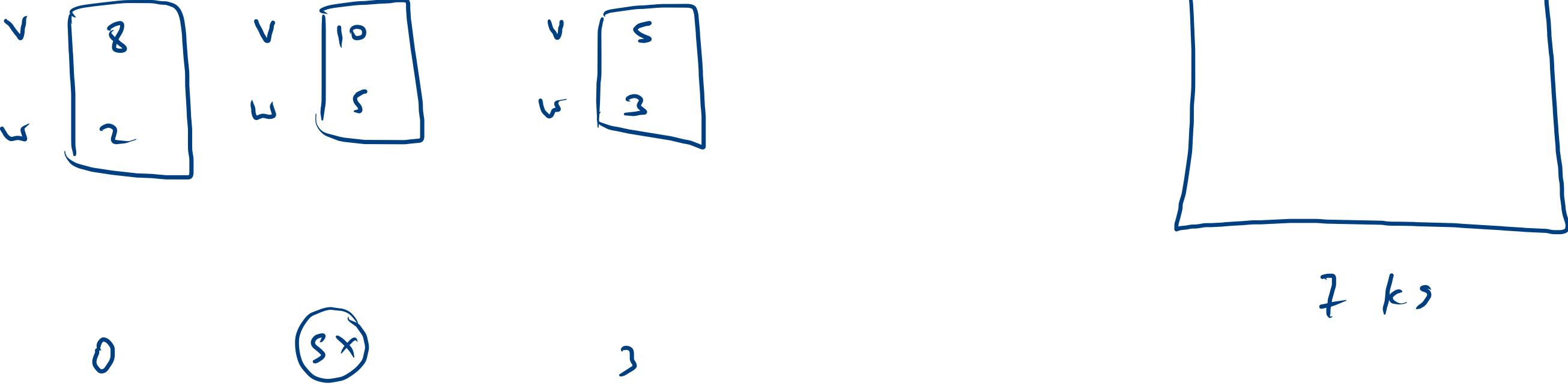
>

s

6

2





v
v
v
v
8
2

v
v
v
10
5

v
v
v
35
3

a b c d

v
v
v
v
v
v

c o r t i

21c)

0	1	2	3	4	5	6	7
0	0	8	8	16	16	27	2^m
$\boxed{8}$ 2	$\boxed{8}$ 2	$\boxed{8}$ 2	$\boxed{8}$ 2	$\boxed{8}$ 2	$\boxed{8}$ 2	$\boxed{8}$ x^3	$\boxed{8}$ x^2

q b
b n
c 2
a d