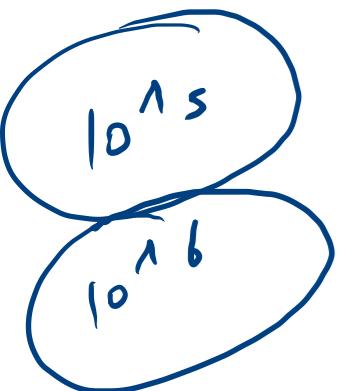
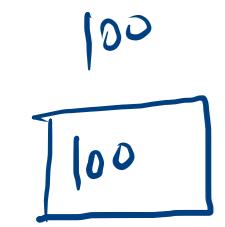
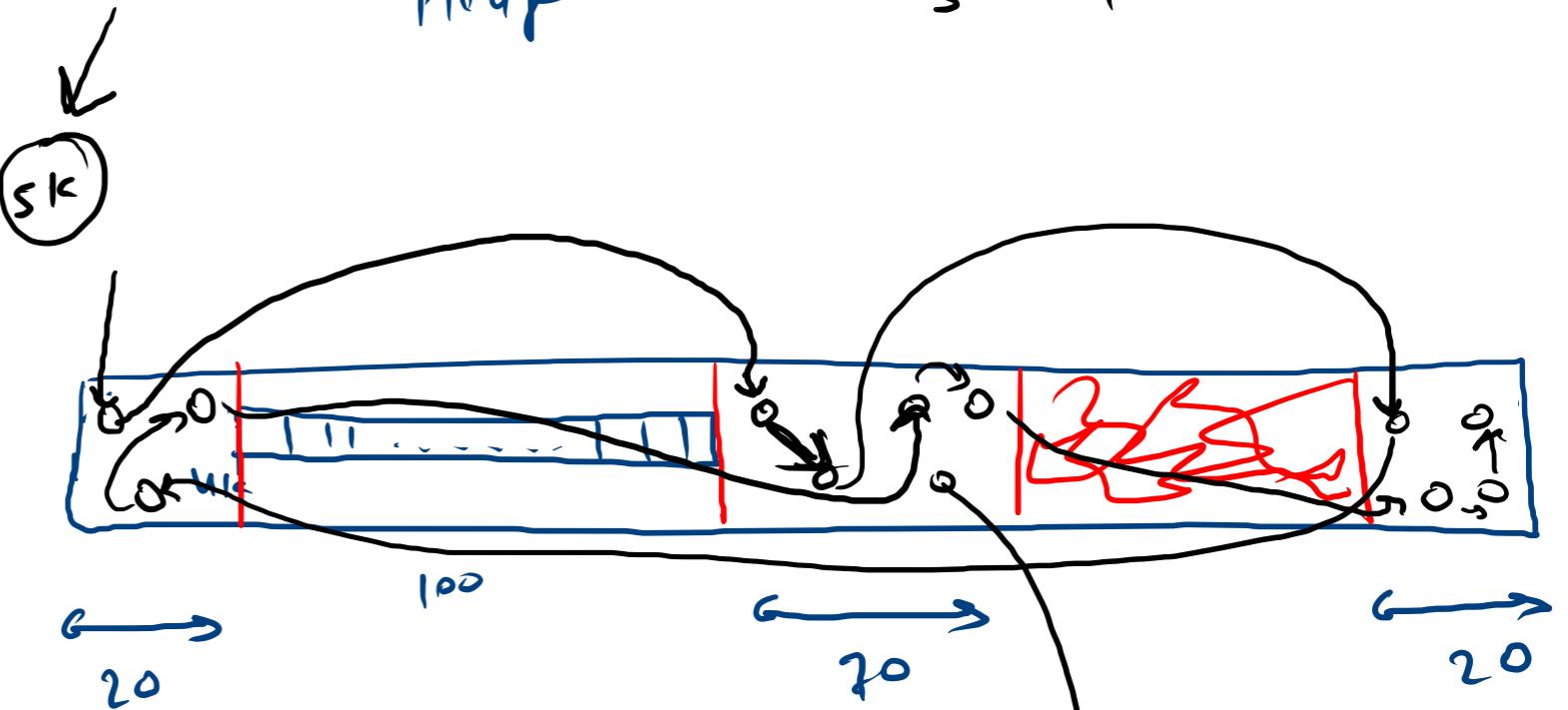


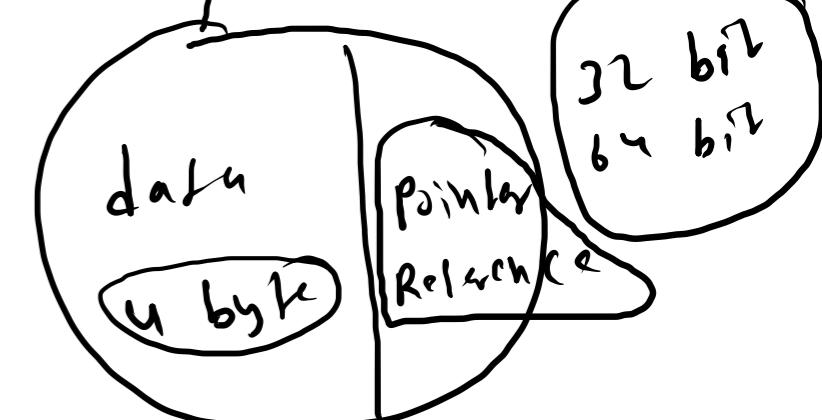
Stack



Heap

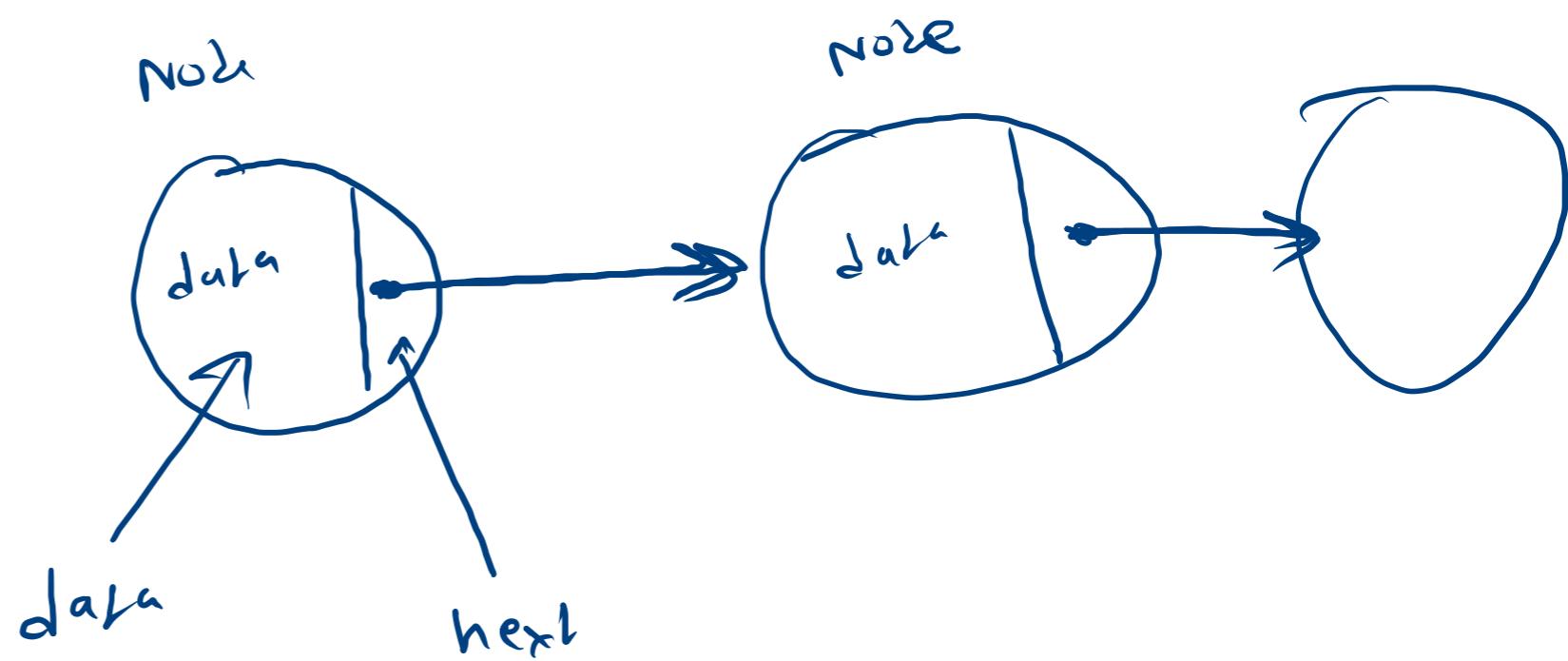


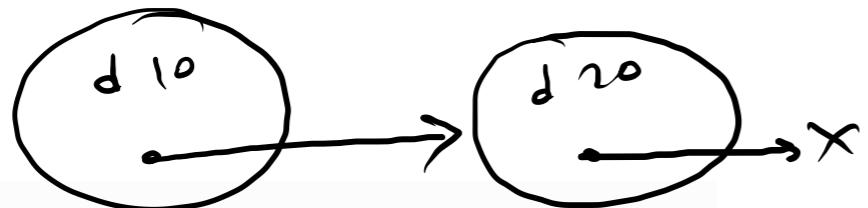
s int



class

```
    Node {  
        int data;  
        Node next; }  
y
```



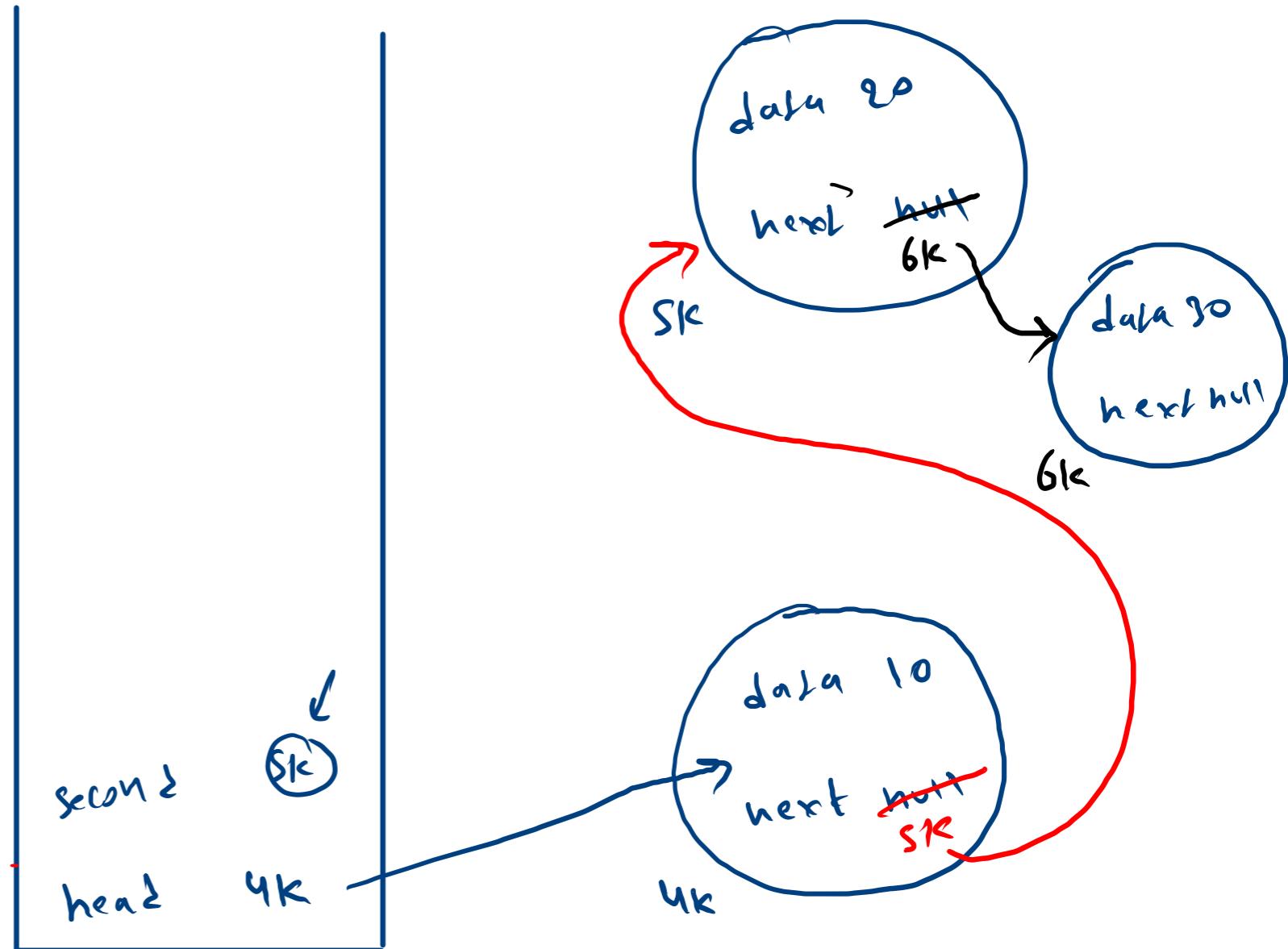


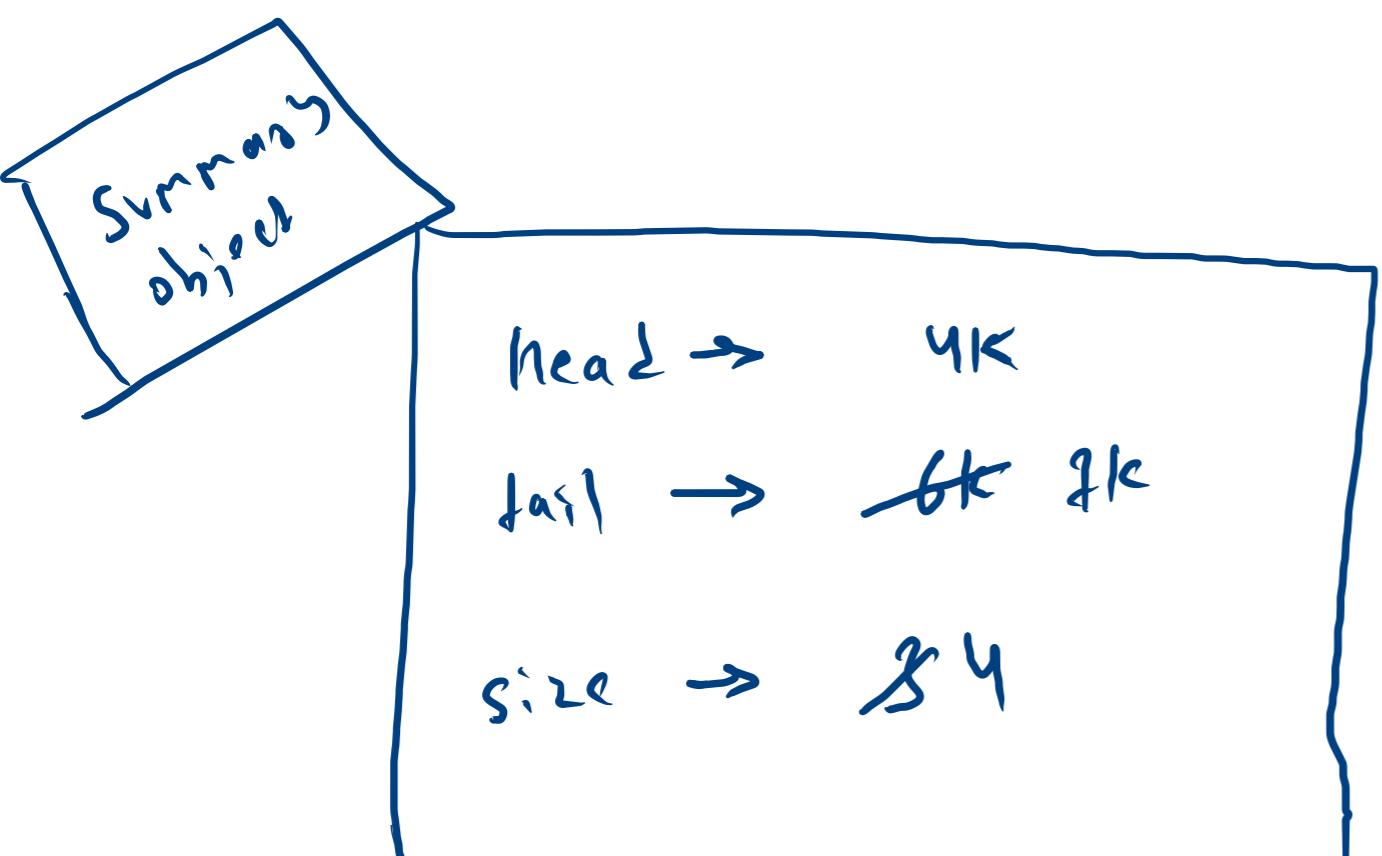
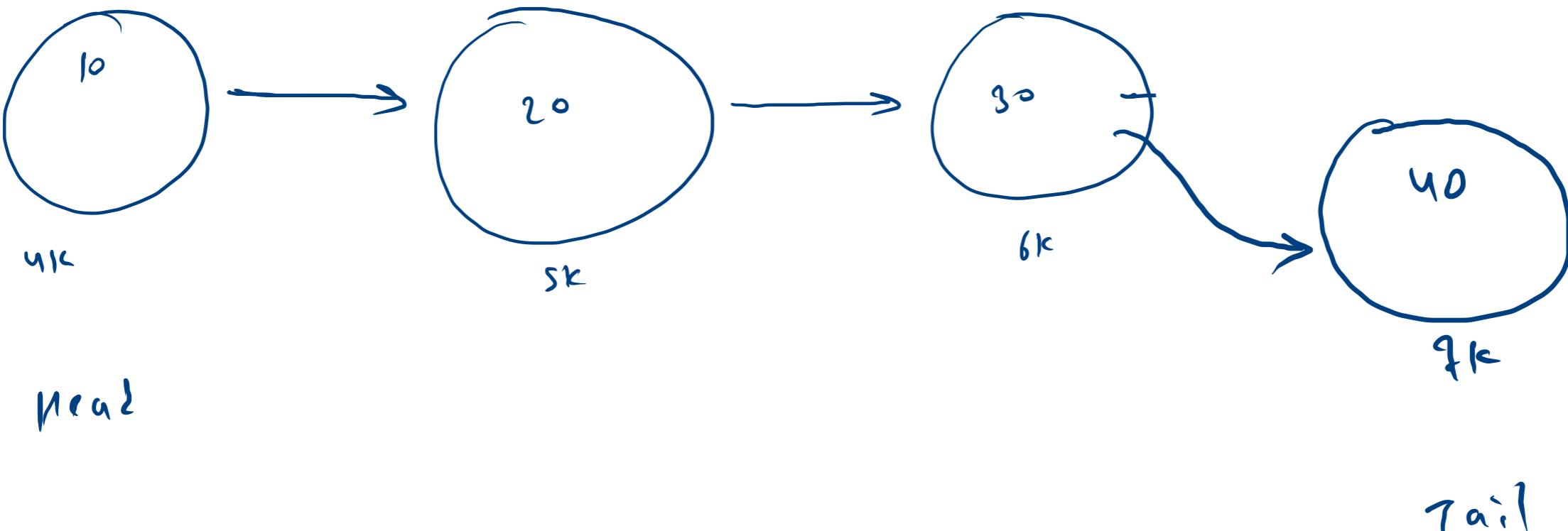
```

public static class Node{
    int data;
    Node next; 10
    Node(int d){
        data = d;
    }
}

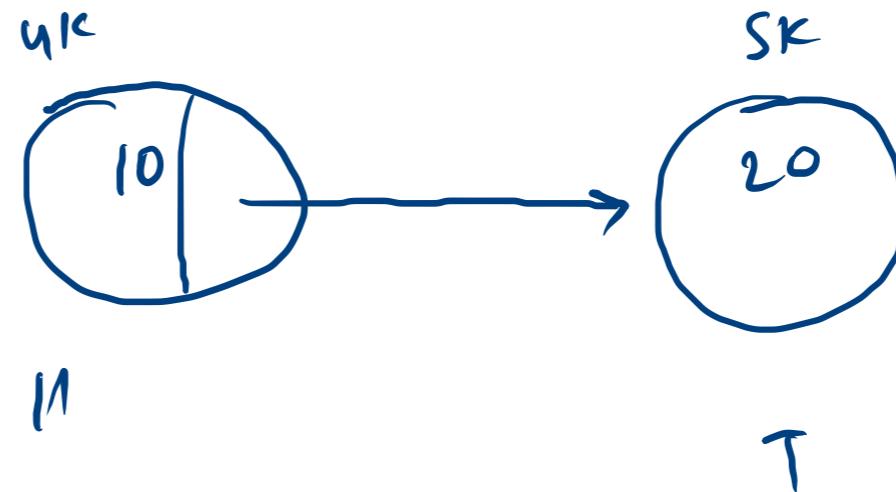
public static void main(String[] args) {
    • Node head = new Node(10);
    • Node second = new Node(20);
    • head.next = second;
}
    |           Sk

```





create new node  
assign data



Summary)

```
public static class LinkedList {  
    Node head;  
    Node tail;  
    int size;  
  
    void addLast(int val) {  
        // Write your code here  
    }  
}
```

---

head	but UK
tail	but SK
size	$8 + 2$

```

public static class Node {
    int data;
    Node next;
}

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

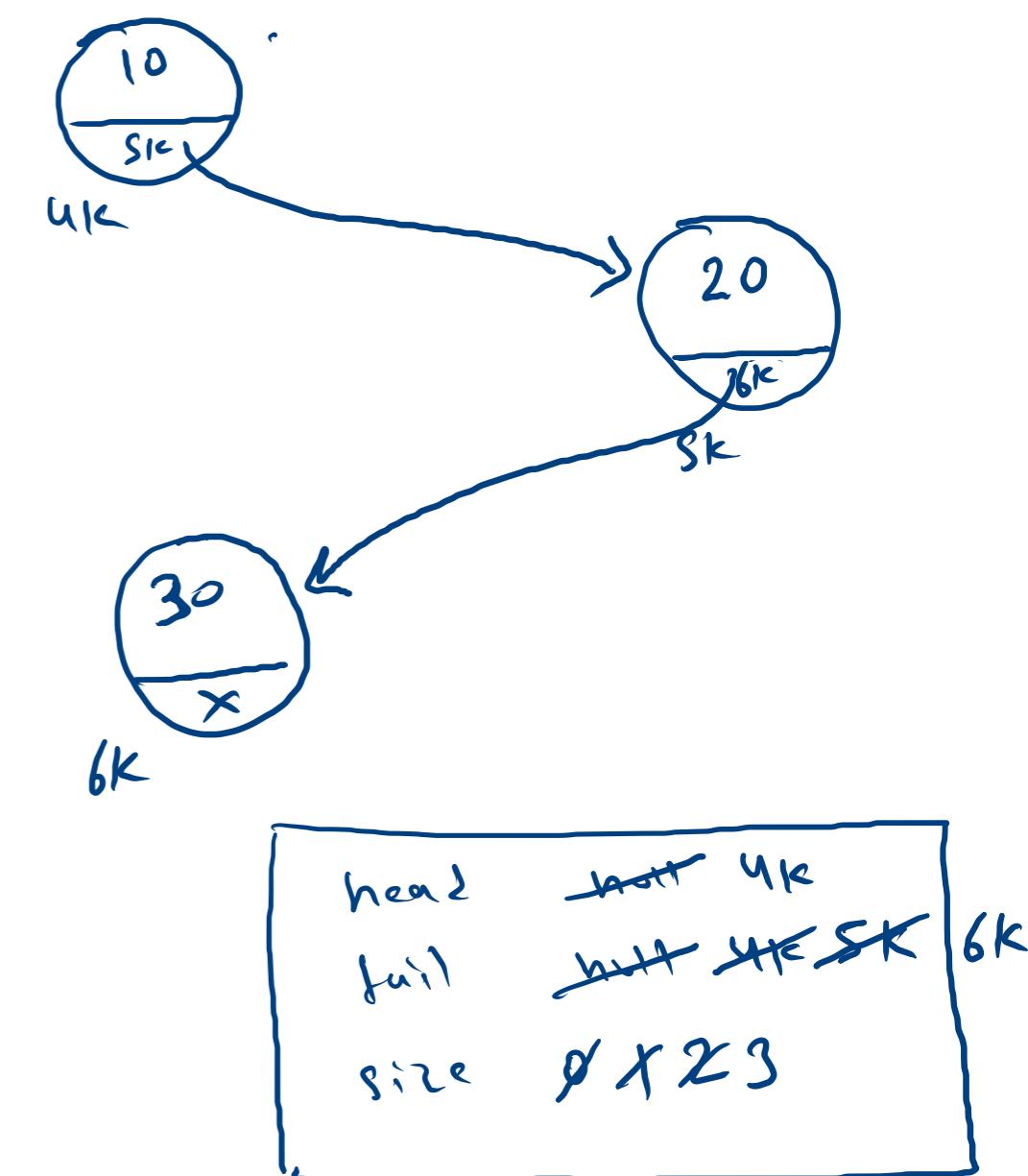
    void addLast(int val) {
        if(size == 0){
            Node node = new Node();
            node.data = val;
            head = node;
            tail = node;
            size = 1;
        }else{
            Node node = new Node();
            node.data = val;
            tail.next = node;
      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

```

~~addLast 10 addLast 20~~

~~addLast 30 addLast 40~~

addLast 50 quit



```

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

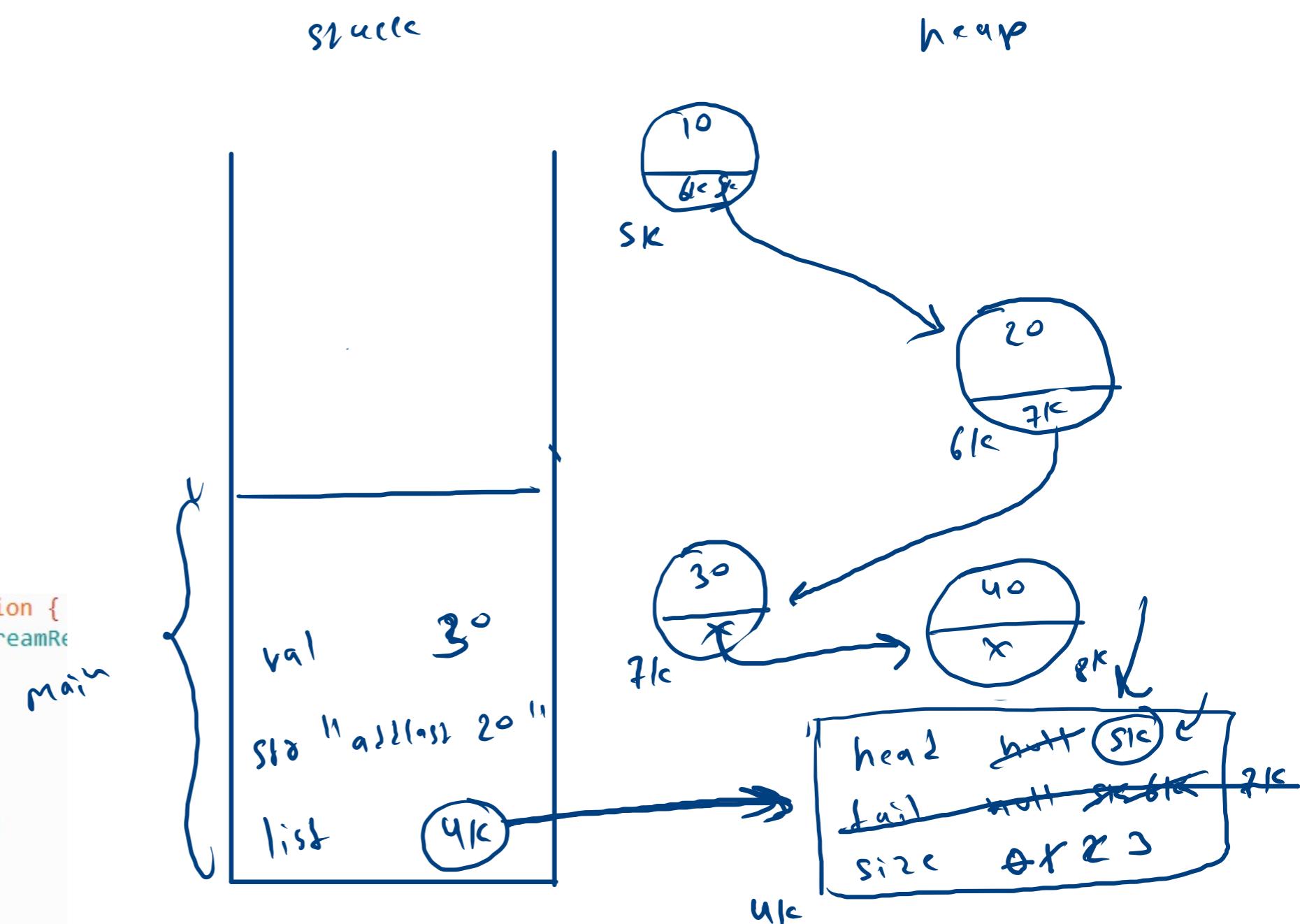
    void addLast(int val) {
        if(size == 0){
            • Node node = new Node();
            • node.data = val;
            • head = node;
            • tail = node;
            size = 1;
        }else{
            • Node node = new Node();
            • node.data = val;
            • tail.next =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

    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(
        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();
        }

        testList(list);
    }
}

```



~~addLast 10~~

~~addLast 20~~

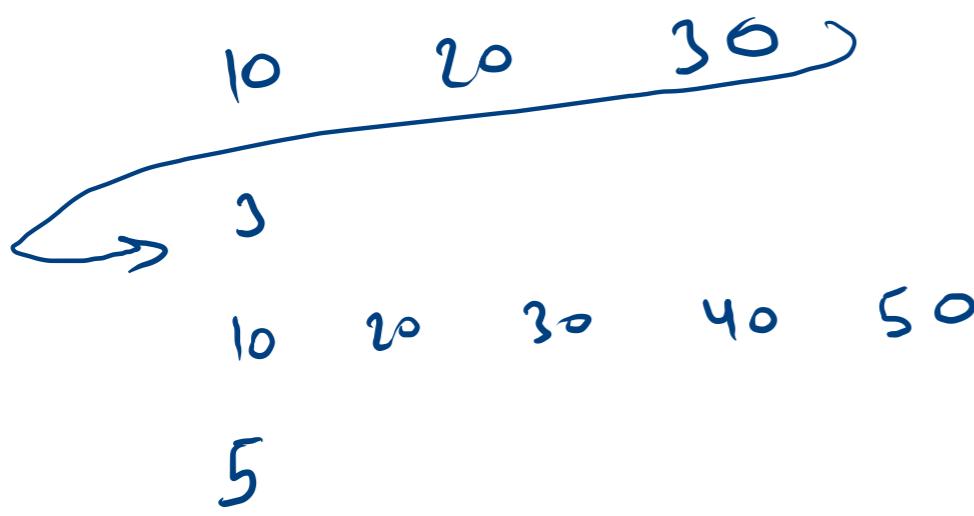
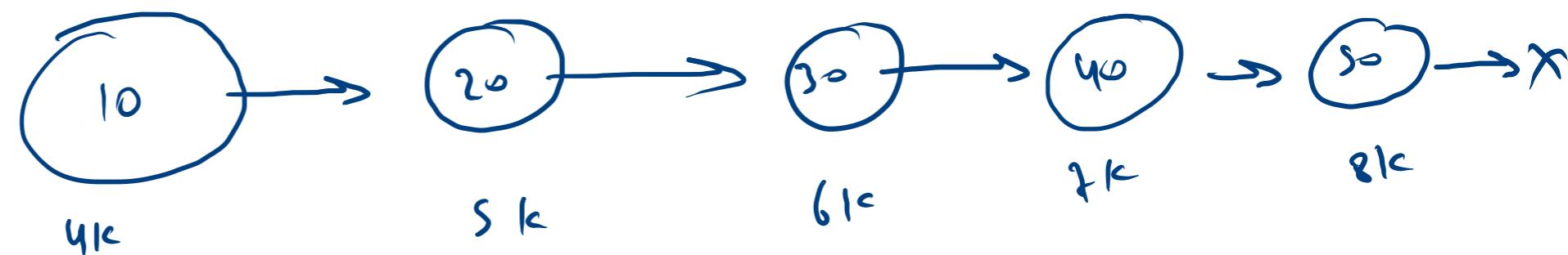
~~addLast 30~~

**display size**

~~addLast 40~~

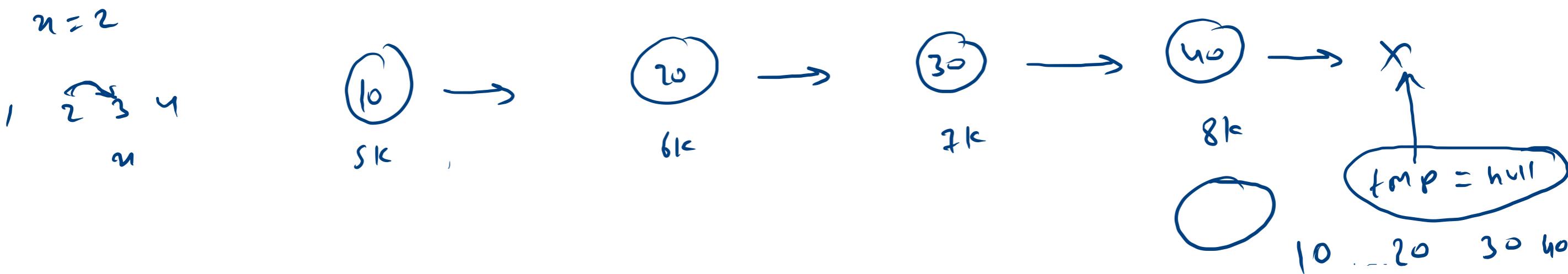
~~addLast 50~~

**display size quit**



head  
tail  
size

\* 41c  
\* 41c \* 61c \* 71c 81c  
0 1 2 4 5

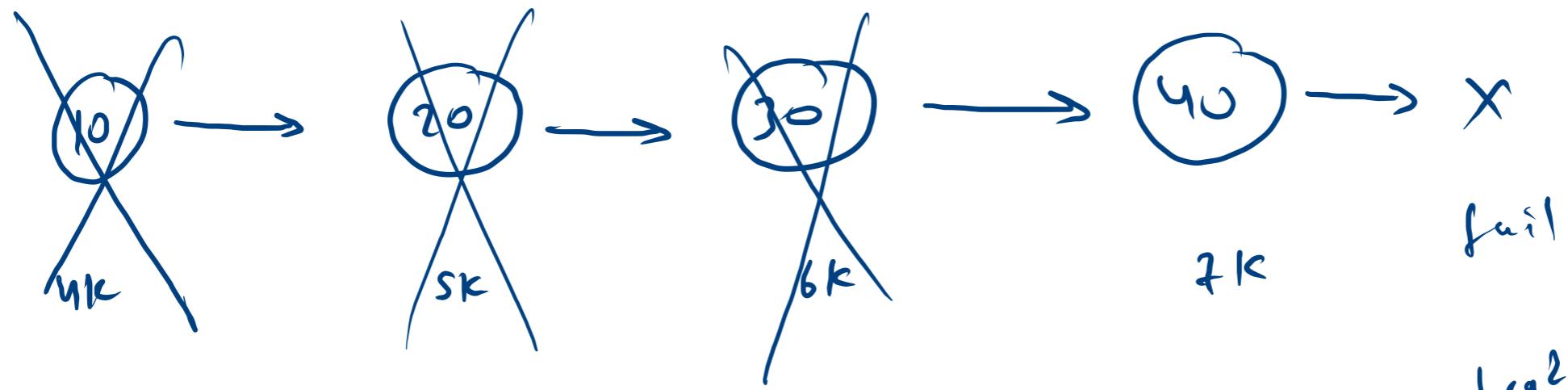


$n = n + 1$   
 $tmp = tmp.next$

Node tmp = head;  
 while (tmp != null) {  
 print (tmp.data);  
 tmp = tmp.next;

y

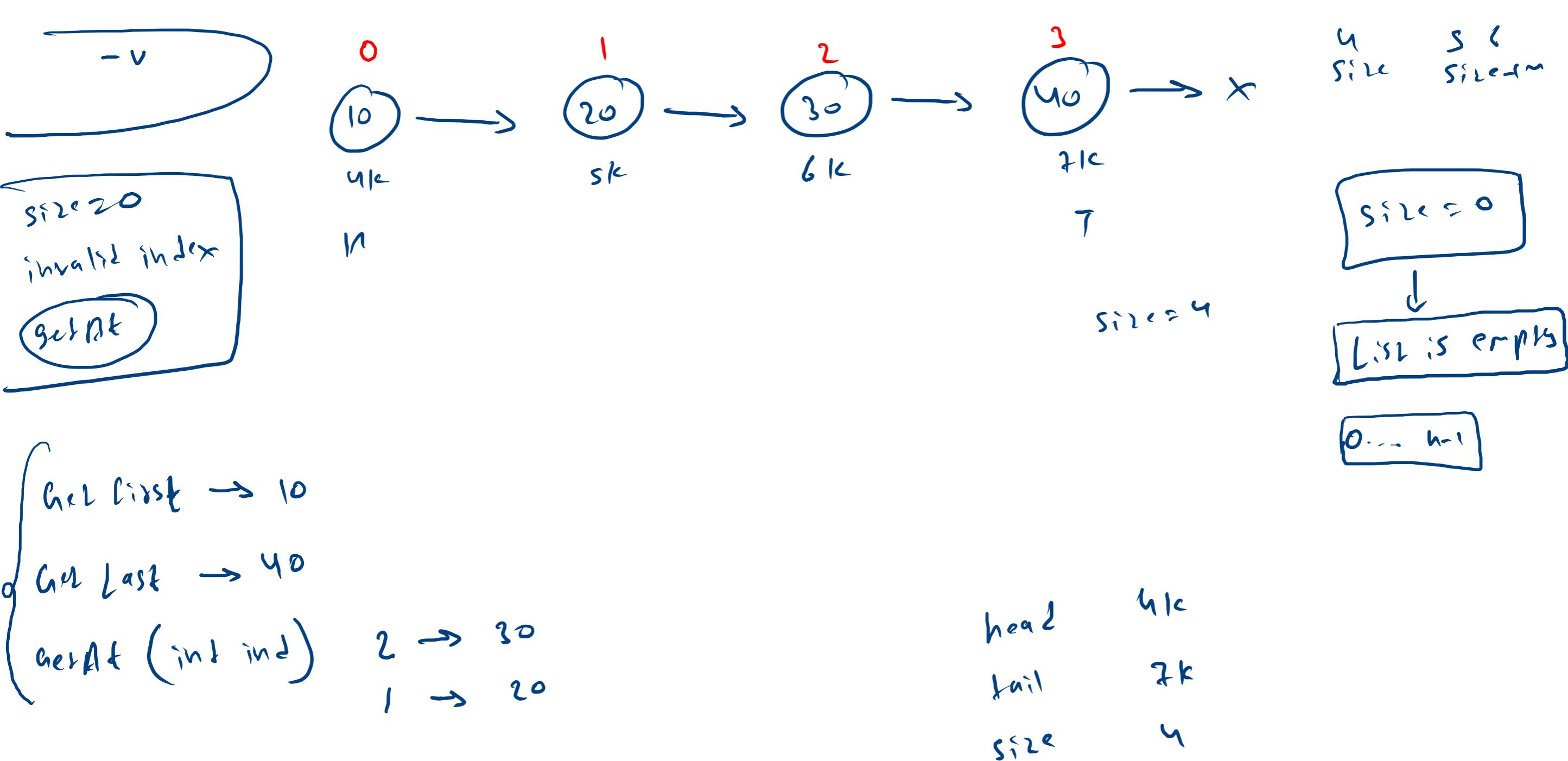
head sk  
 tail 8k  
 size 4

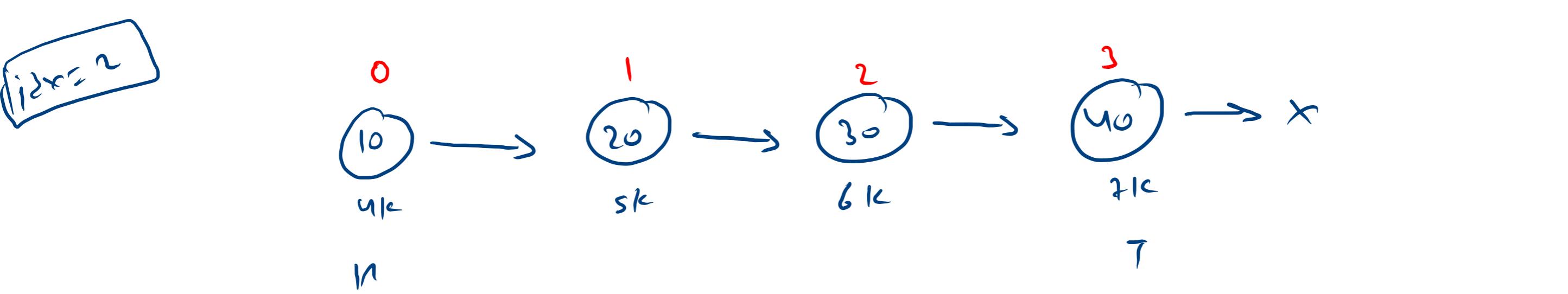


head <sup>2</sup>	X
tail <sup>1</sup>	X
size	0

removeFirst

head	yK	sk	6K	2K	X
tail				2K	X
size	X	8	2	X	0





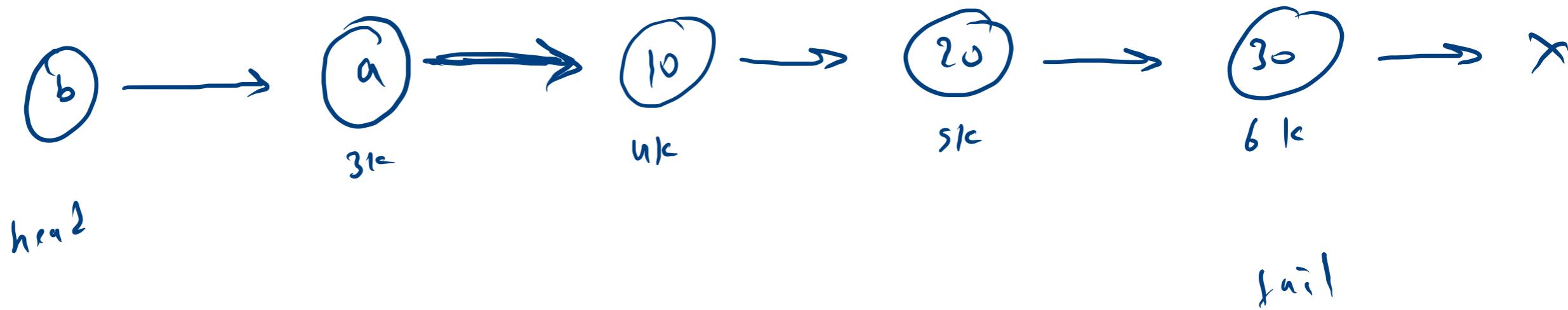
$i$

0      1      2      tmp

$\text{int } i = 0;$   
 $\text{Node } \text{tmp} = \text{head};$   
 $\text{while } (i < i \geq n - 2)$

$\{\text{tmp} = \text{tmp}. \text{next};$   
 $i++\}$

3



node new node

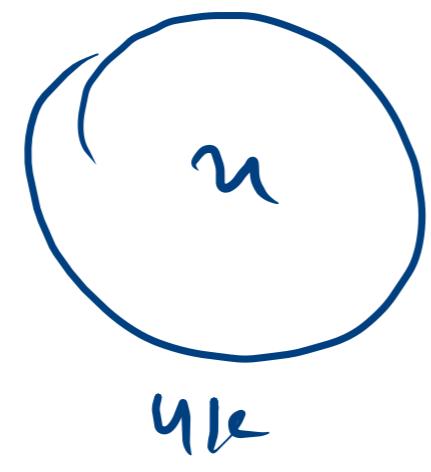
node.data = val;

node.next = head

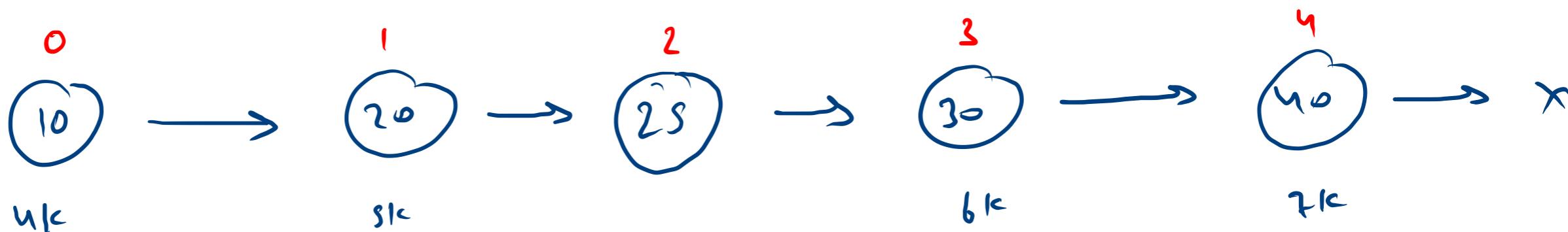
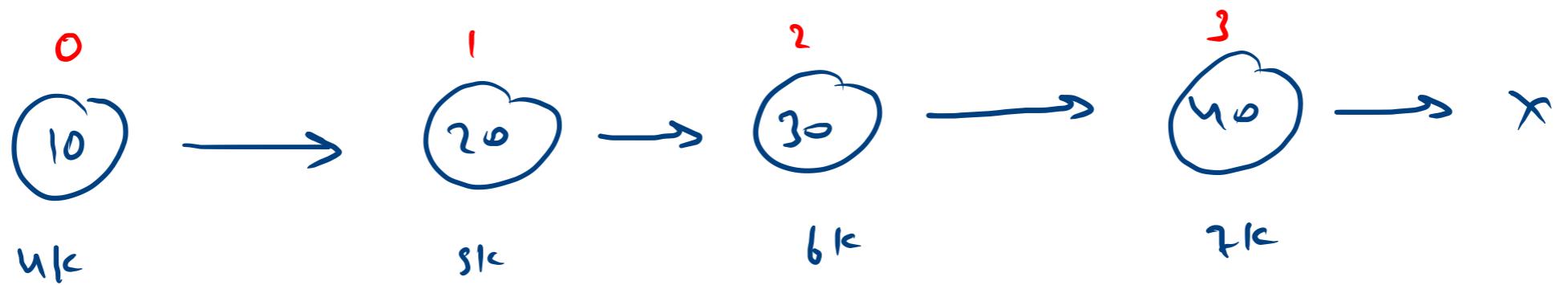
head = node

size++;

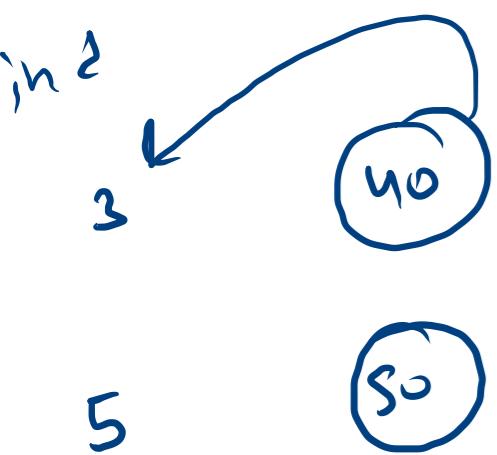
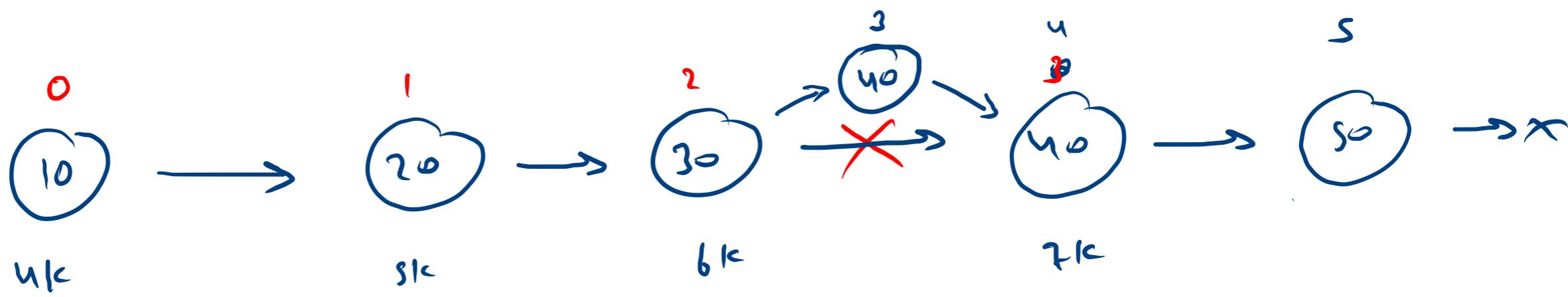
head	41c	31e
tail	61k	
size	245	



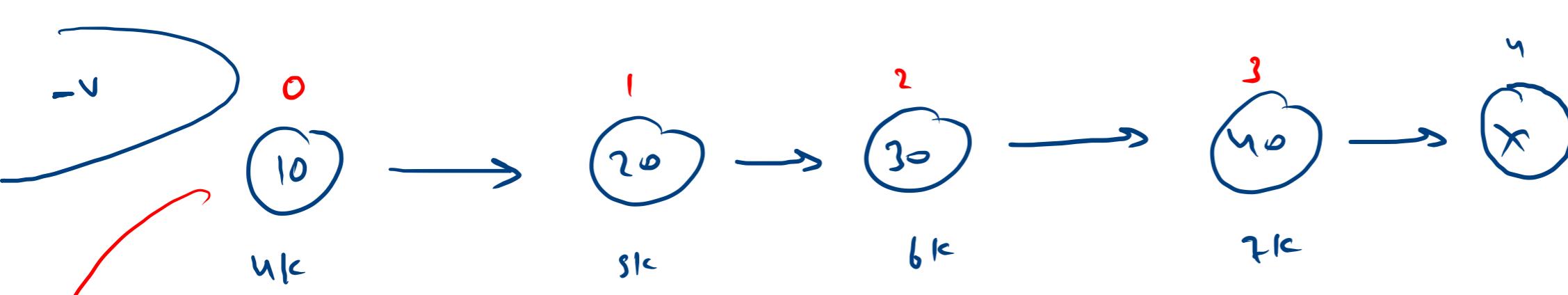
head = ~~huff~~ 4k  
last = ~~huff~~ 4k  
size = Ø 1



head = 41c  
tail = 71c  
size = 4

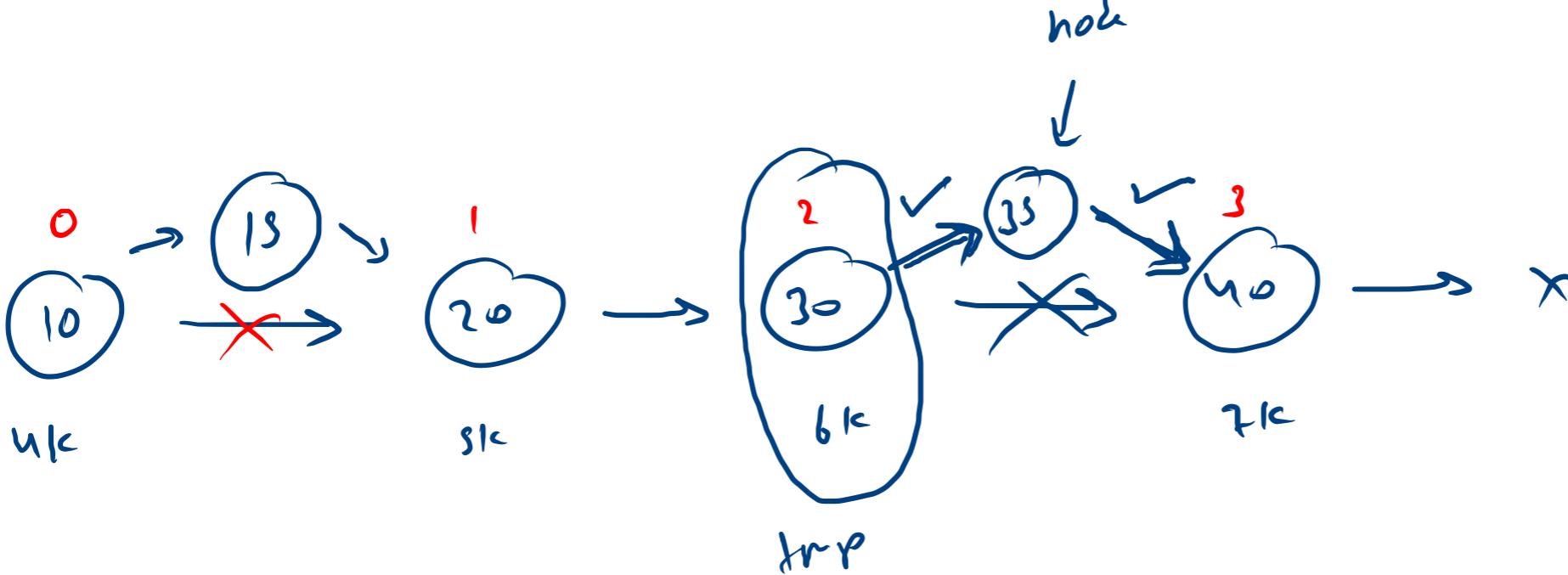


$size = 4$   
 $tail = 8k$   
 $head = 4k$



$$ind = 3$$

`tmp` at index  
 $(ind-1)$

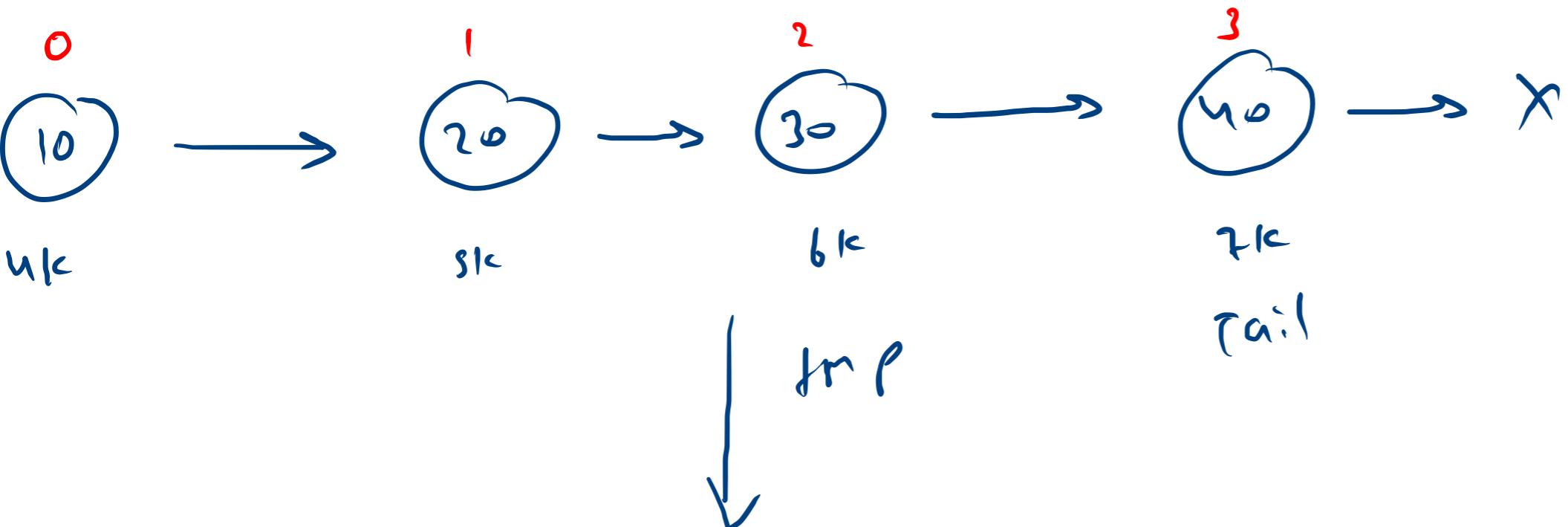


$$\text{node.next} = \text{tmp.next}$$

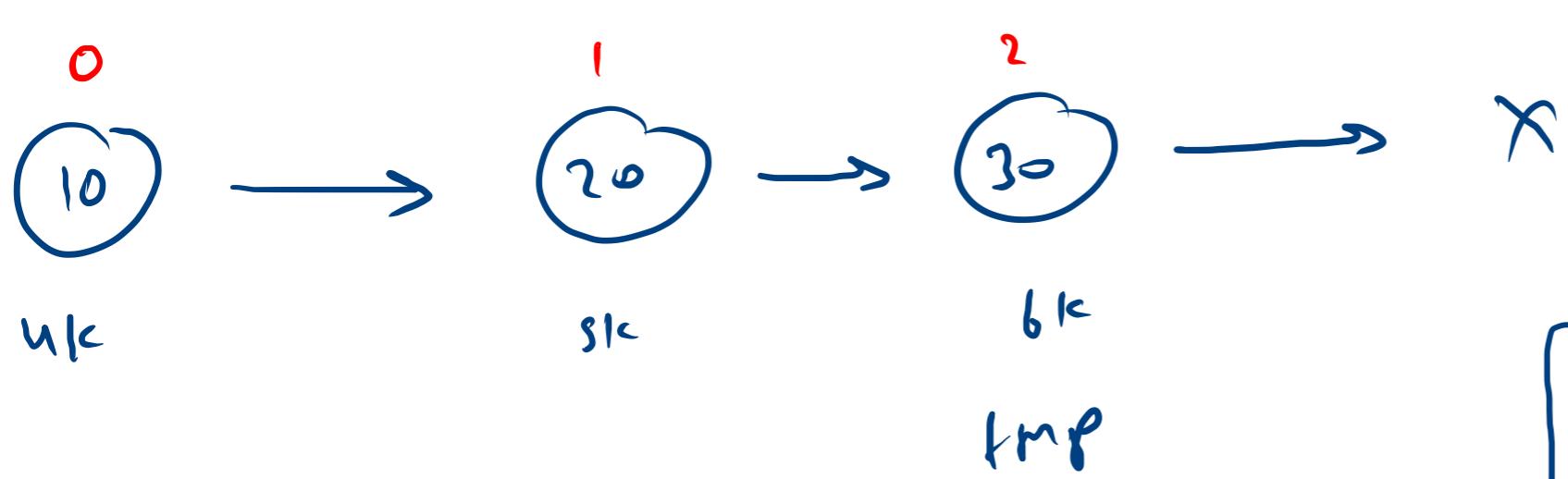
$$\text{tmp.next} = \text{node}$$

$$ind = 3, 15$$

$$ind = 1, 15$$



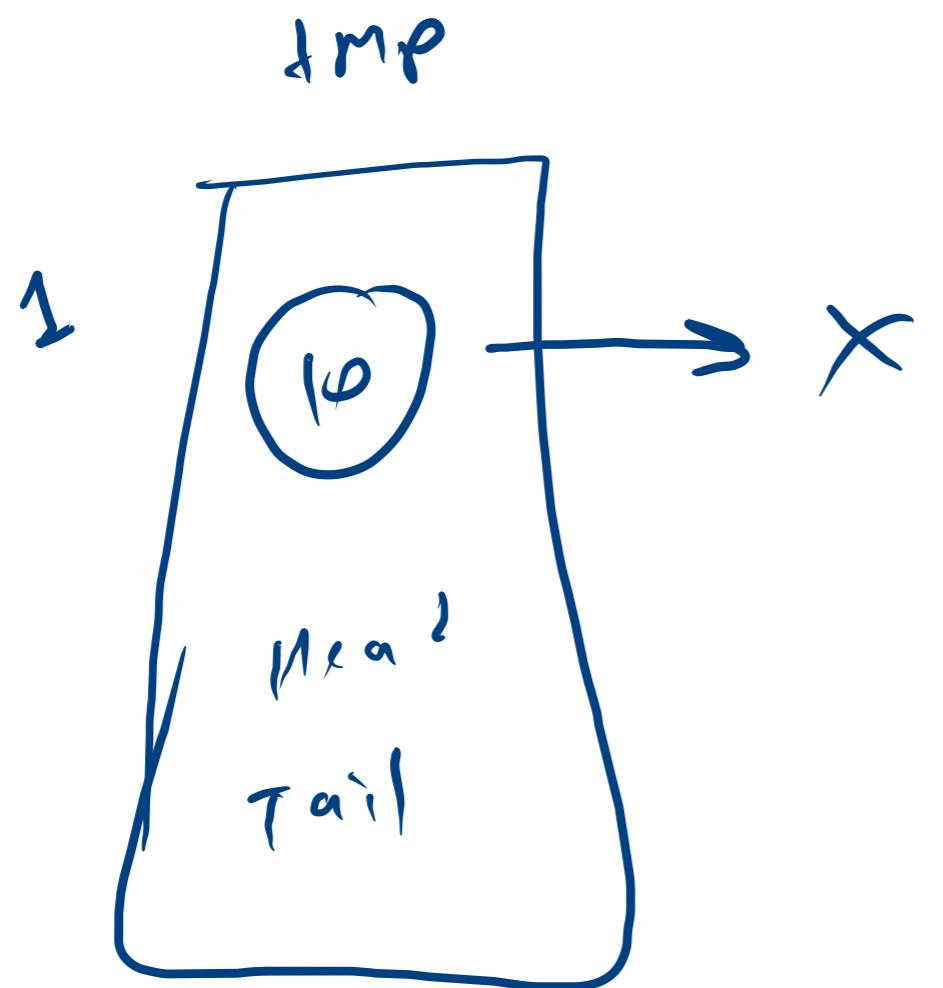
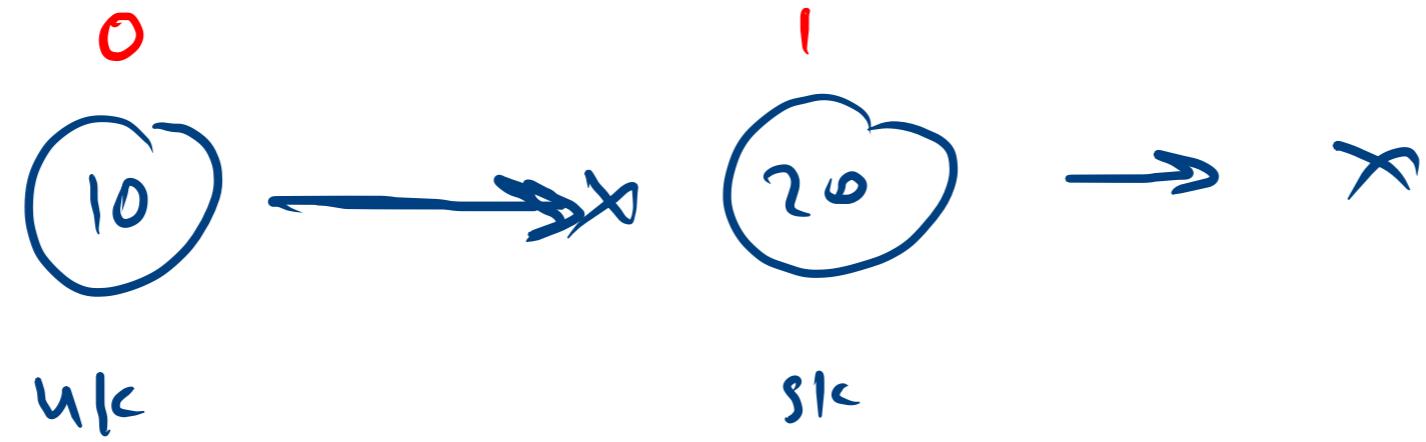
$head = 41c$   
 $tail = f1c$   
 $size = 4$

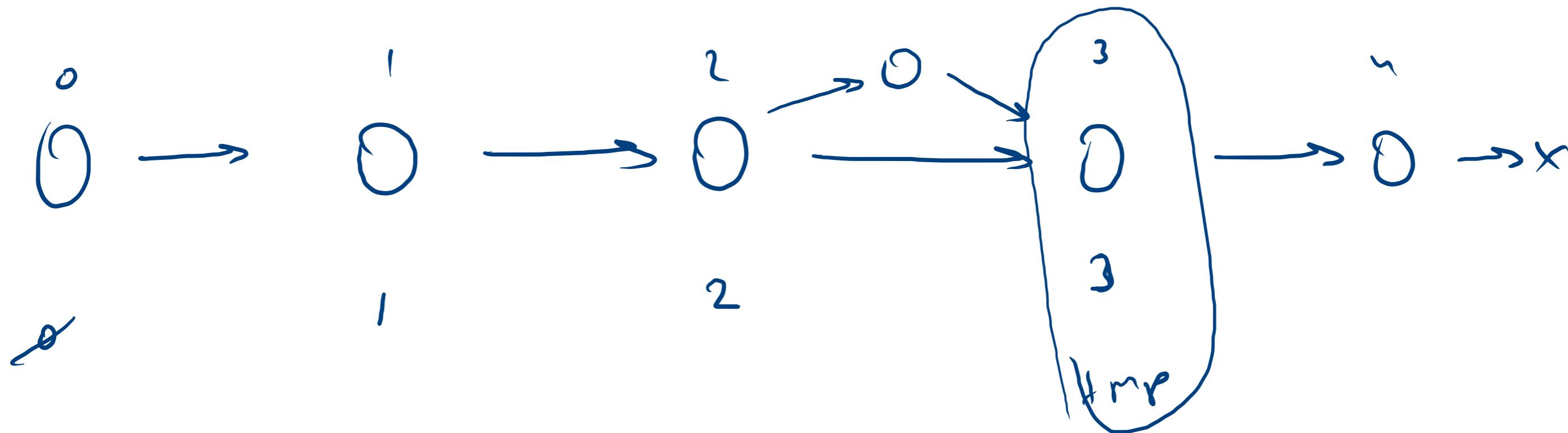


$head = 41c$   
 $tail = b1c$   
 $size = 3$

$tmp.next = tail$

\* while  $tmp.next \neq tail$





$i < 3$   
 $i < 3$   
 $(i < \text{idx}) \wedge$   
 $i = \text{idx}$   
 $i > 3$

$\text{tmp} = \text{tmp}.\text{next}$

$i++$