

List ll = new ArrayList

Classname	var	new class name
P	obj	new P();
P	obj	new C();
C	obj	new P();
C	obj	new C();

main

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    // Case-1
    P obj = new P();
    System.out.println(obj.d); // 1
    System.out.println(obj.d1); // 10
    obj.fun(); // P
    obj.fun1(); // P
    // Case -2
    P obj = new C();
}
    
```

Stacks = 2

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    // Case-1
    P obj = new P();
    System.out.println(obj.d); // 1
    System.out.println(obj.d1); // 10
    obj.fun(); // P
    obj.fun1(); // P
    // Case -2
    P obj = new C();
    System.out.println(obj.d);
    System.out.println(obj.d1); // 10
    System.out.println(obj.d2);
}
    
```

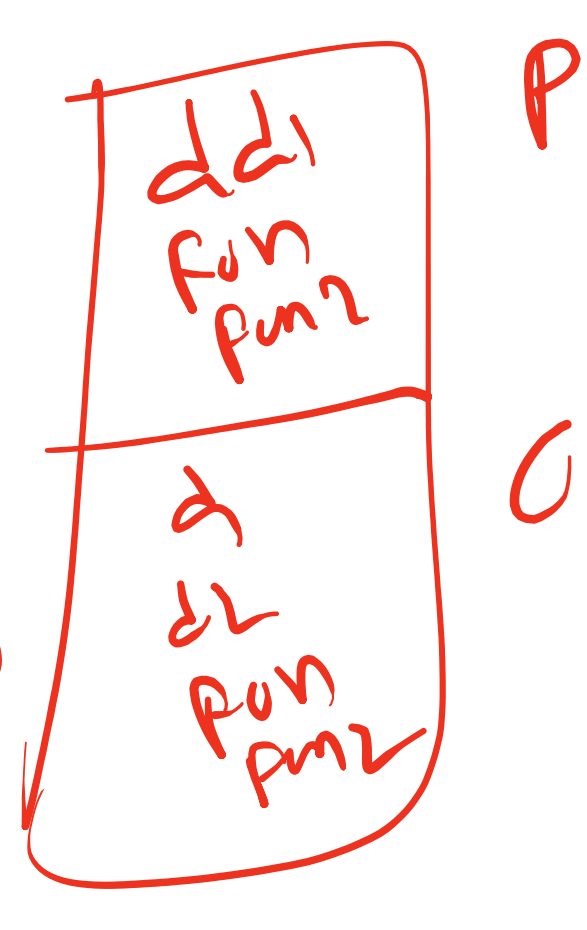
P obj = new C();

List<Integer> ll = new ArrayList<>()

ll.add()

```

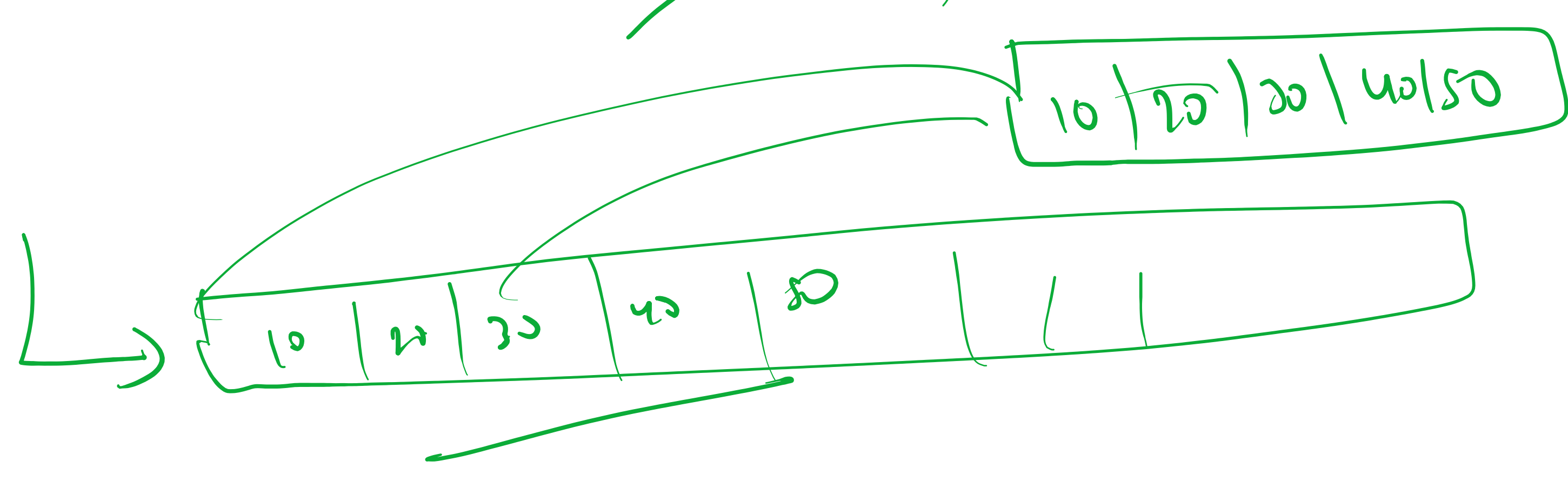
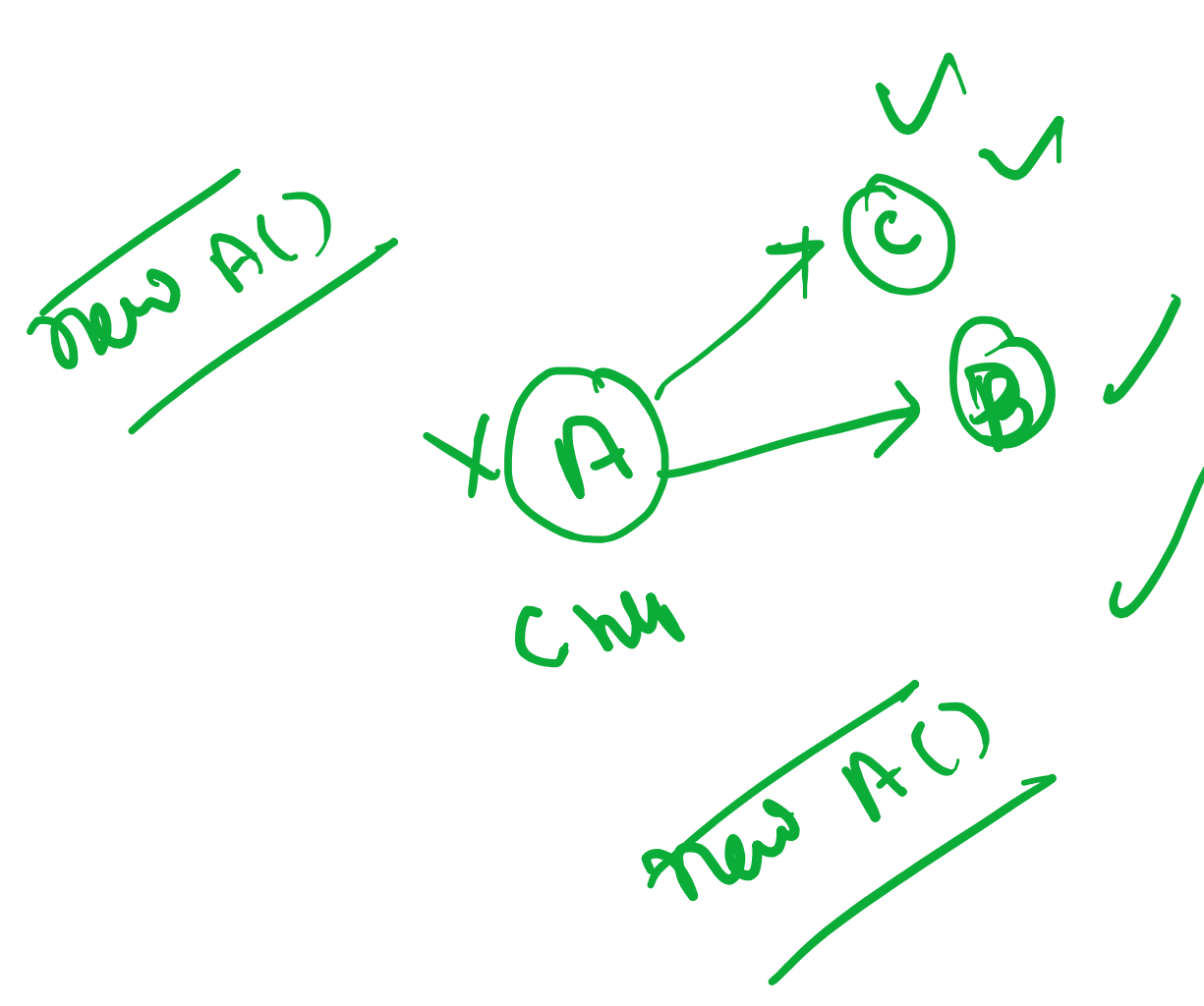
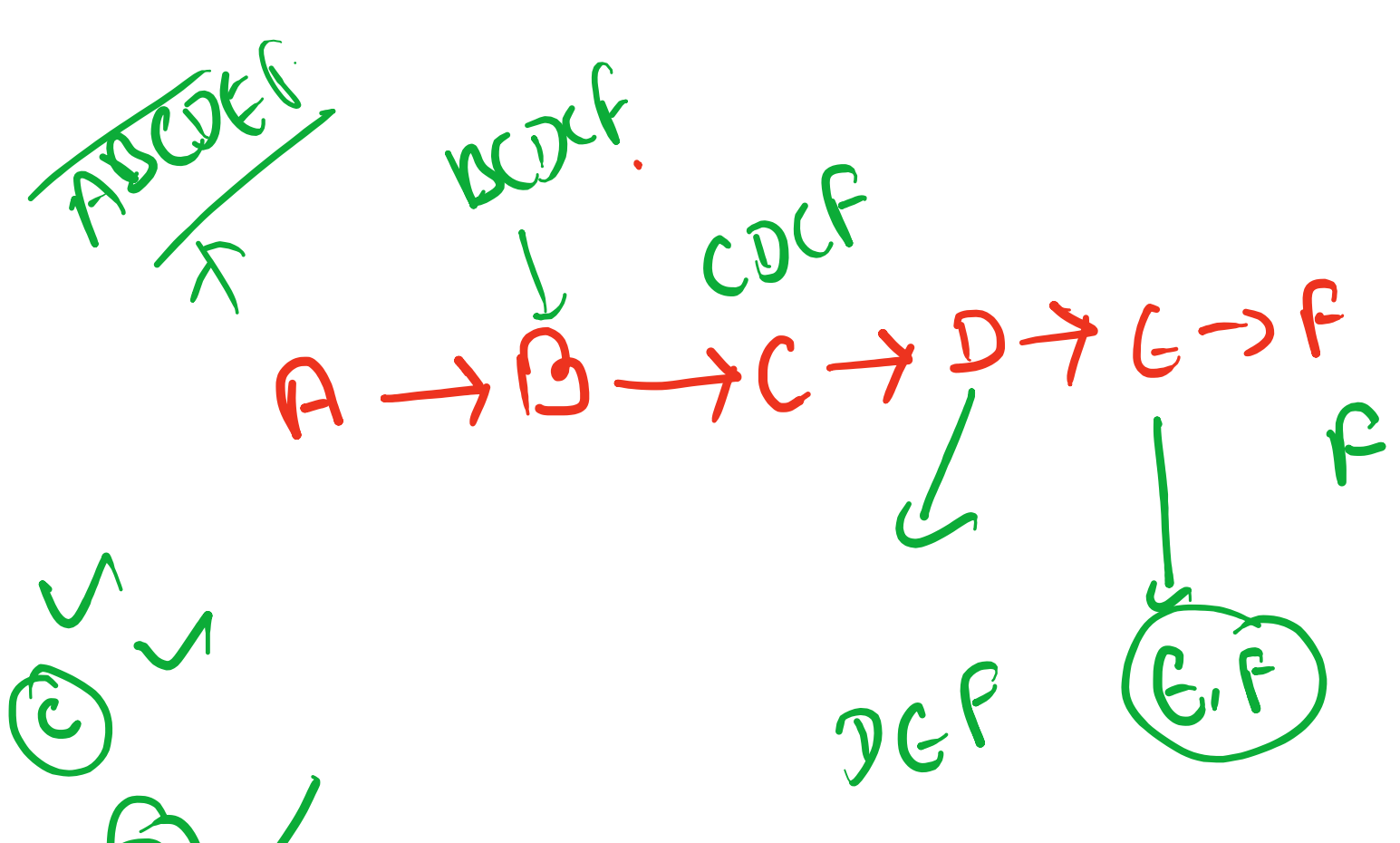
P obj = new C();
System.out.println(obj.d);
System.out.println(obj.d1); // 10
System.out.println(((C) obj).d2);
System.out.println(((C) obj).d);
obj.fun(); // P
obj.fun1(); // P
((C) obj).fun2(); // C
    
```



d=1
d1=10

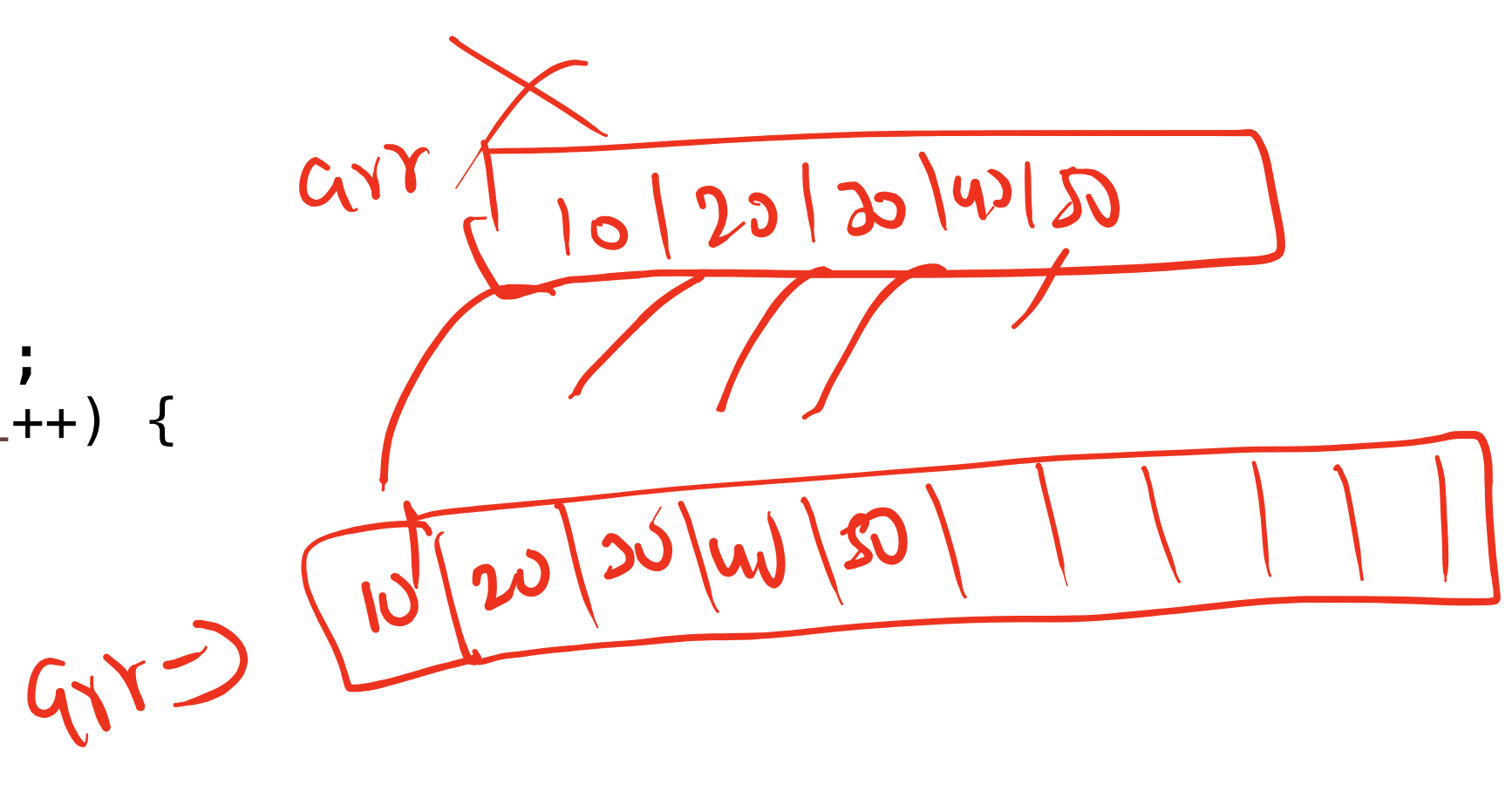
C obj = new P();

Multi Level



```

@Override
public void push(int item) {
    if (isfull()) {
        int[] a = new int[2 * arr.length];
        for (int i = 0; i < arr.length; i++) {
            a[i] = arr[i];
        }
        arr = a;
    }
}
    
```

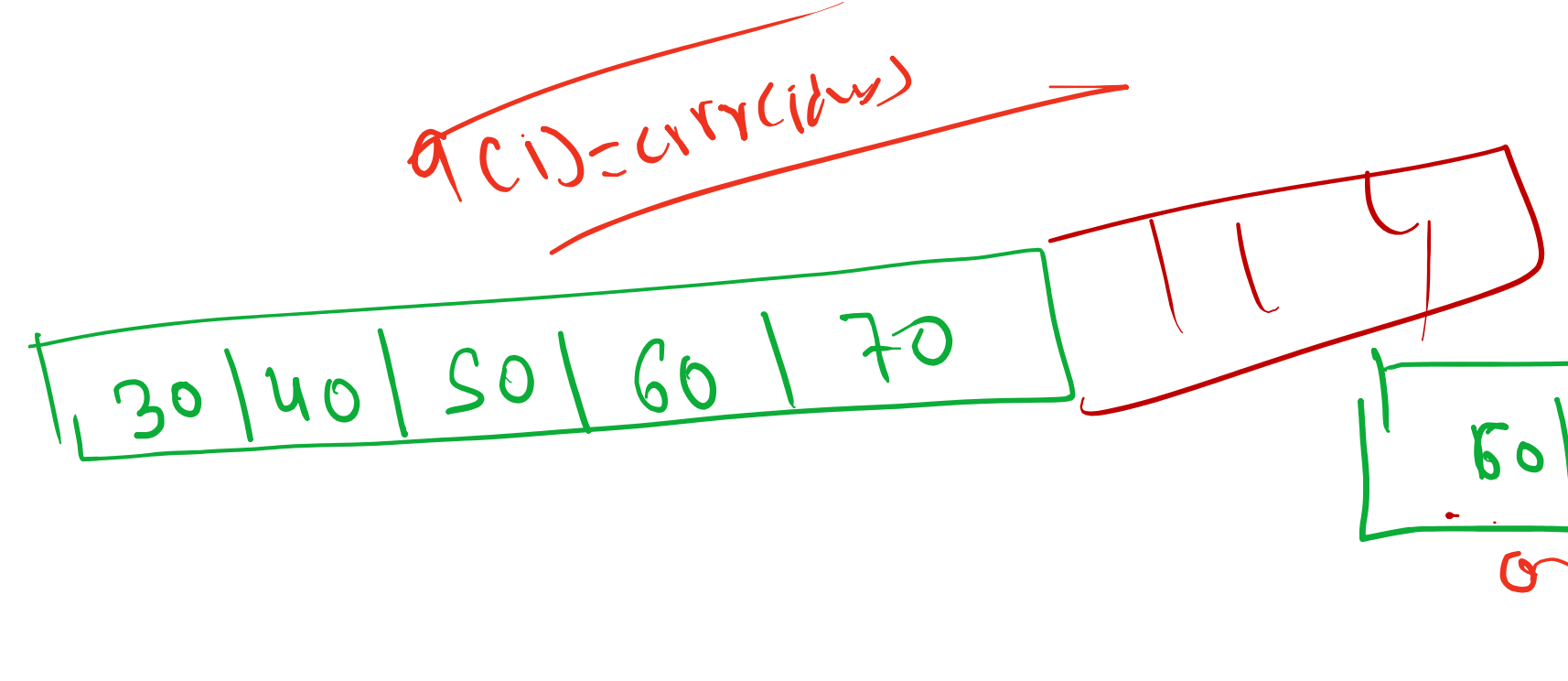


```

public class Dynamic_Stack extends Stack {
    @Override
    public void push(int item) {
        if (isfull()) {
            int[] a = new int[2 * arr.length];
            for (int i = 0; i < arr.length; i++) {
                a[i] = arr[i];
            }
            arr = a;
        }
        super.push(item);
    }
    public static void main(String[] args) {
        Dynamic_Stack ds = new Dynamic_Stack();
        ds.push(10);
        ds.push(20);
        ds.push(30);
        ds.push(40);
        ds.push(50);
        ds.push(60);
    }
}
    
```

idx = 1
idx = 2
idx = 3
idx = 4
idx = 5

arr[10] = 20; arr[20] = 30; arr[30] = 40; arr[40] = 50; arr[50] = 60;



```

for (int i = 0; i < arr.length; i++) {
    int idx = (front+i)%arr.length;
}
    
```

i=0 (2+0)%5=2
i=1 (2+1)%5=3
i=2 (2+2)%5=4
i=3 (2+3)%5=0



arr[0] = arr[2]
arr[1] = arr[3]
arr[2] = arr[4]
arr[3] = arr[0]

Stack
push arr
pop arr

