ArrayList

Switch Case

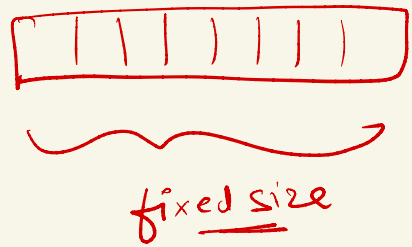
Ternary Operator

By: RITIK RAMUKA

Arraylist

```
int[] arr = new int[10];
```

Array of a fixed size 10.

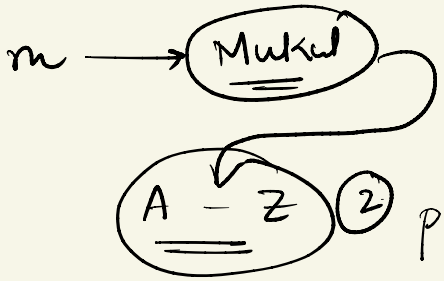


60 students

array size 60

a new student joined ?

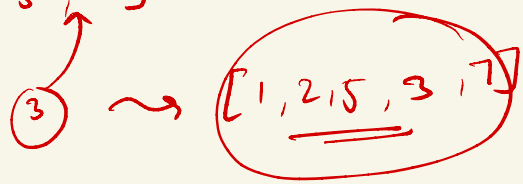
You can occupy (1) that student.



place in b/w the array?

Not possible.

arr = [1, 2, 5, 7]



Arrays

→ A continuous chunk of memory

→ fixed size (static)

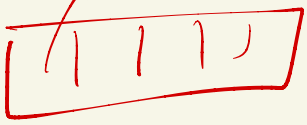
→ primitive data types, Generics
- → Wrapper classes
- → Strings
- → Collections

```
int[] arr = new int[10];
char[] bar = new char[10];
float[]
```

class A

```
Integer[] arr = new Integer[10];
String[] arr = new String[10];
A[] arr = new A[10];
```

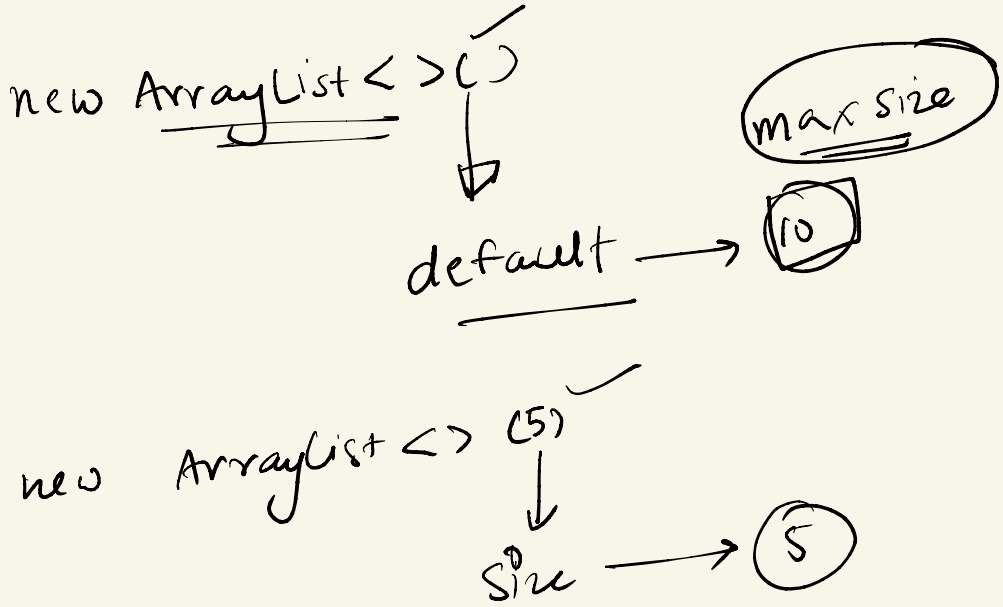
obj
A

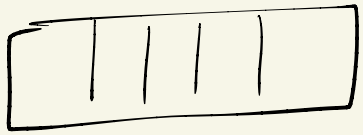


Arraylist

1. Continuous Memory
2. Size is variable (dynamic array)
3. Generics ⟶ Wrapper classes
   ⟶ String
   ⟶ Collections.

* We can't make arraylist of primitive datatype.


new ArrayList<>()
max size
default
10
new ArrayList<>(5)
size
5



add (6)

ArrayList<Integer> list = new ArrayList();

size = 10

↳ capacity

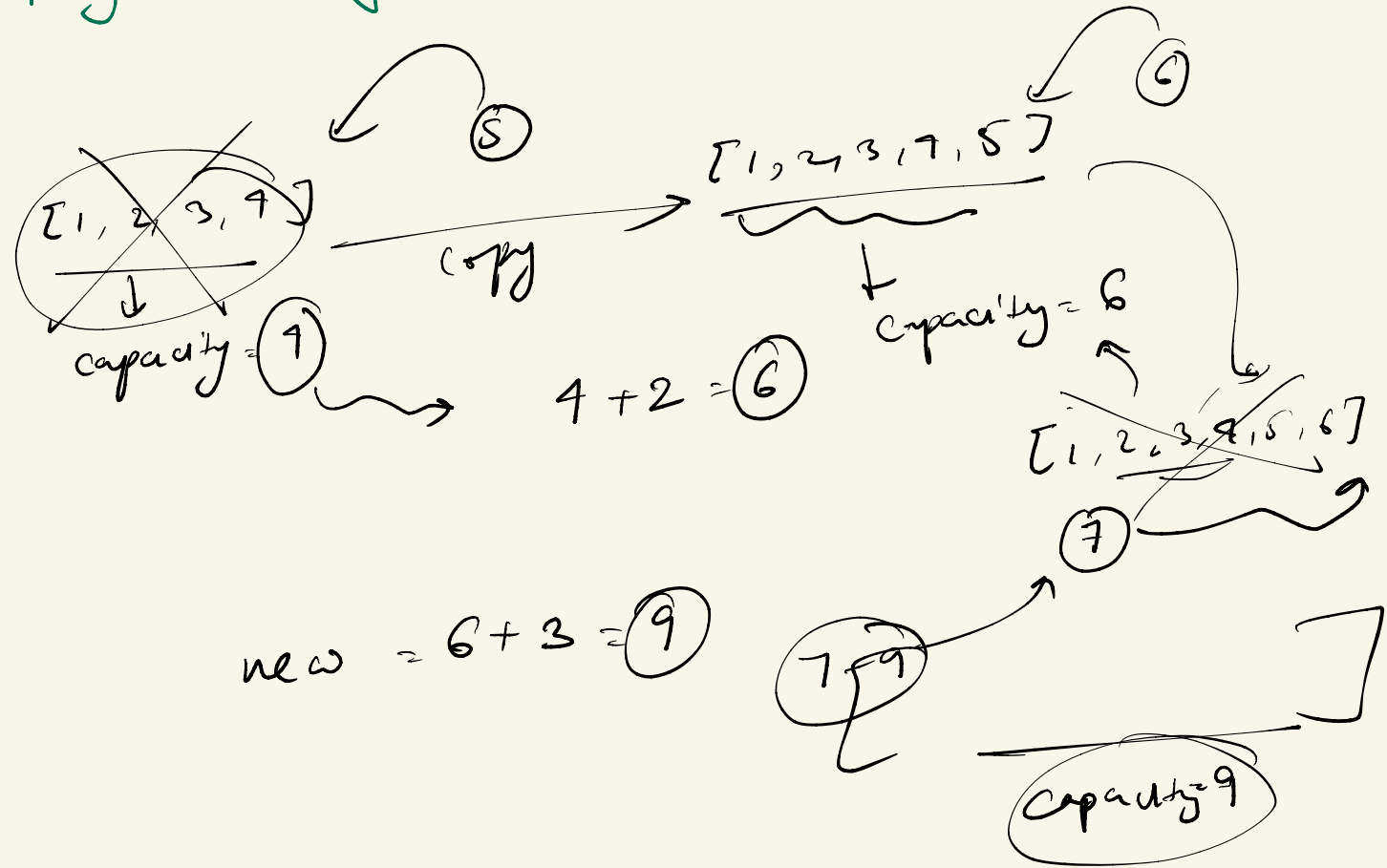
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,

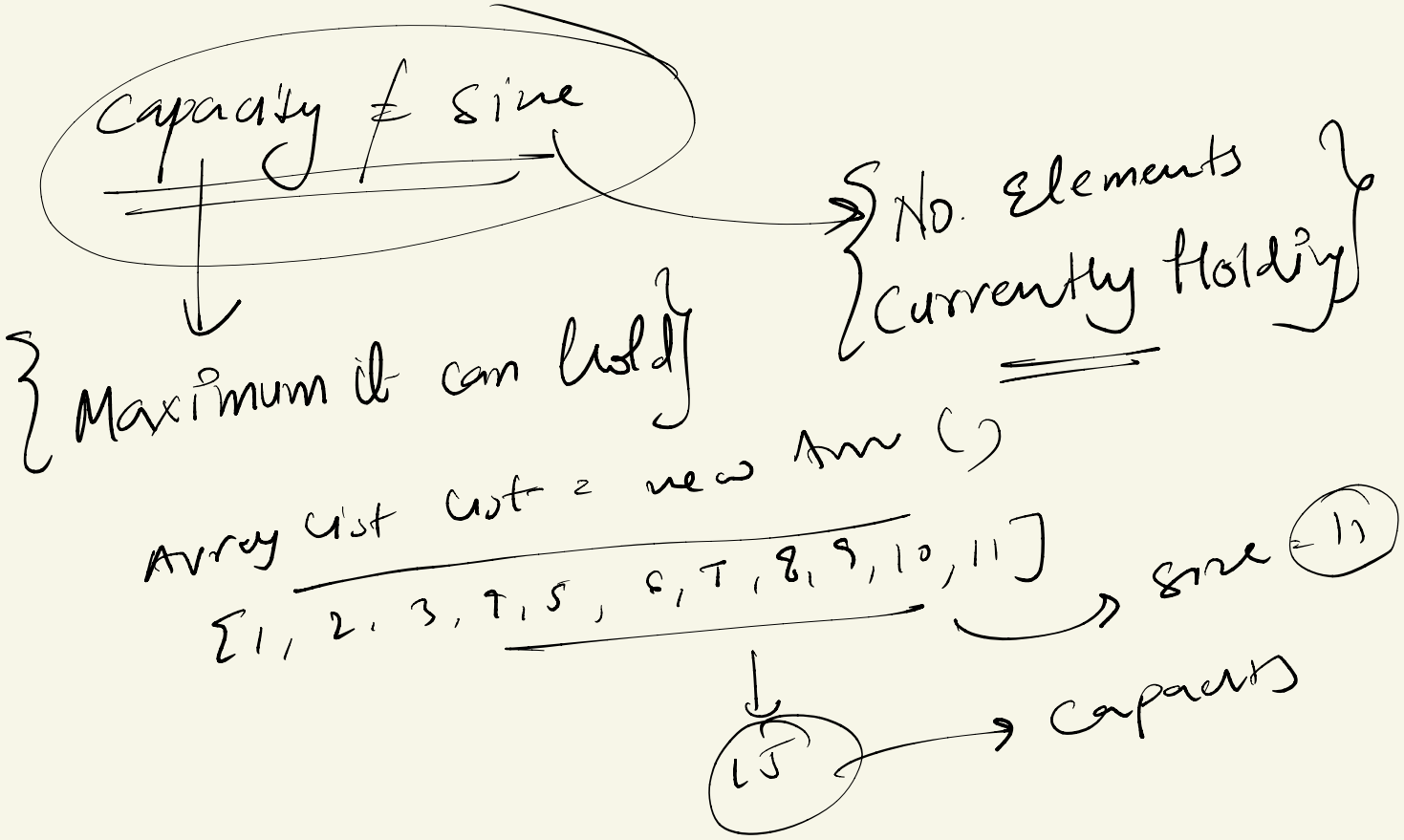
13, 14, 15, 16 22

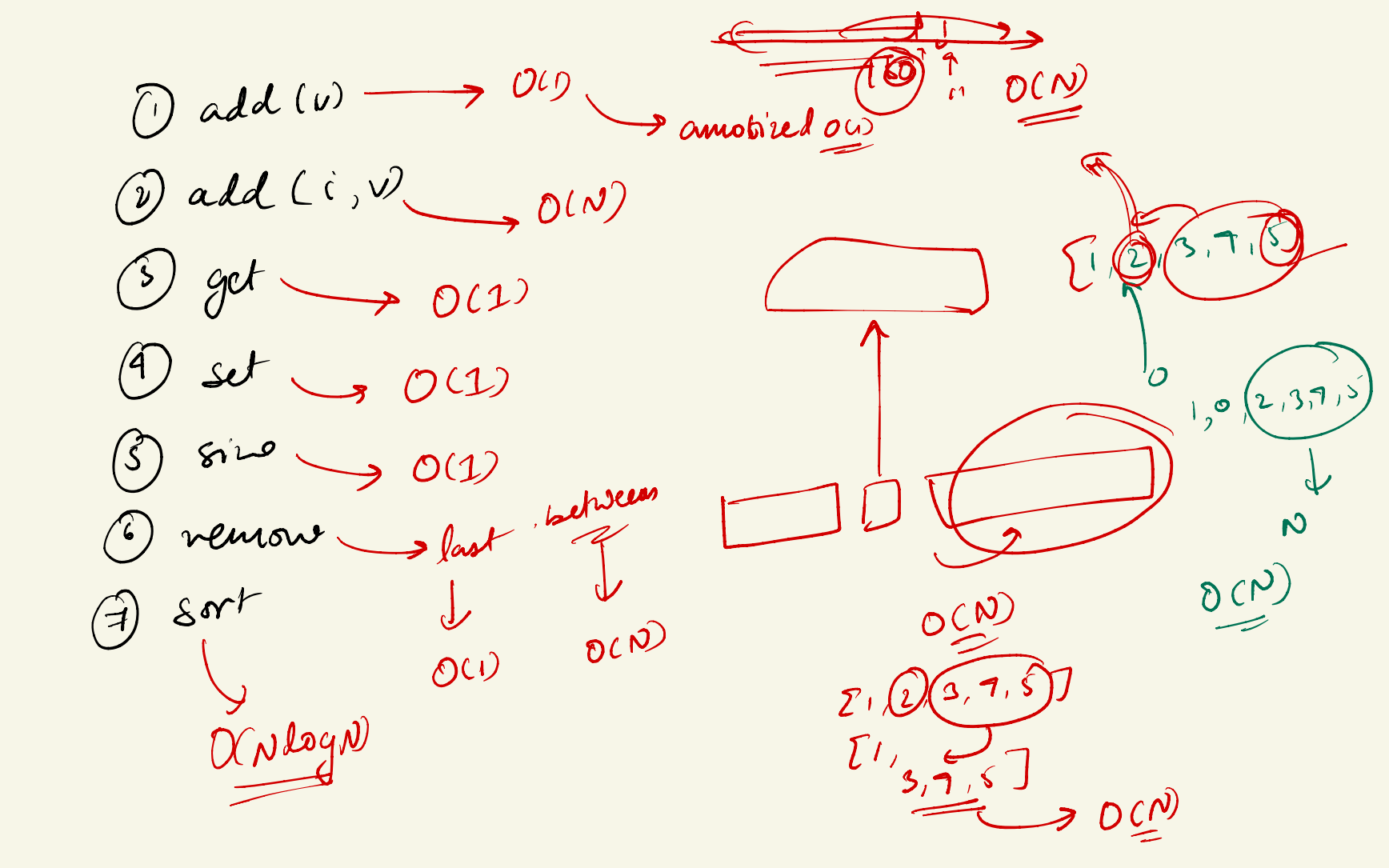
capacity = 10

capacity = 15

15 + 7 = 22


ArrayList<Integer> list = new ArrayList(4);
5
[1, 2, 3, 4]
capacity = 4
copy
[1, 2, 3, 4, 5]
6
capacity = 6
4 + 2 = 6
[1, 2, 3, 4, 5, 6]
7
new = 6 + 3 = 9
7 9
capacity 9


Capacity ≠ size
No. elements currently Holding
Maximum it can hold
ArrayList list = new Arr()
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
size 11
15
capacity

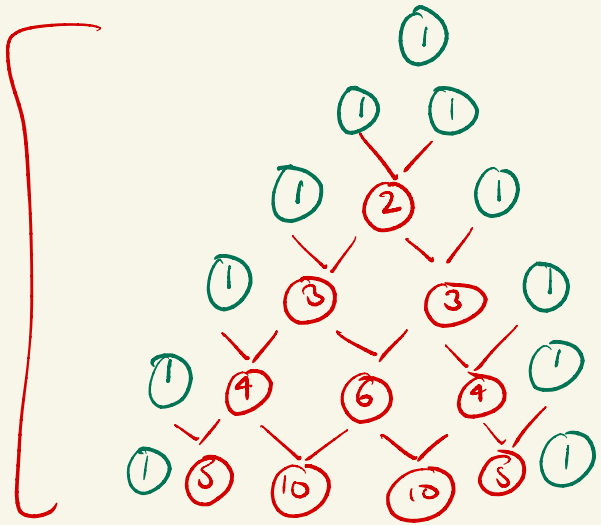

1 add (v) → O(1) → amortized O(1)
2 add (i, v) → O(N)
3 get → O(1)
4 set → O(1)
5 size → O(1)
6 remove → last → O(1)
between → O(N)
7 sort → O(N log N)
O(N)
O(N)
O(N)
O(N)
O(N)

Q. Composite Number

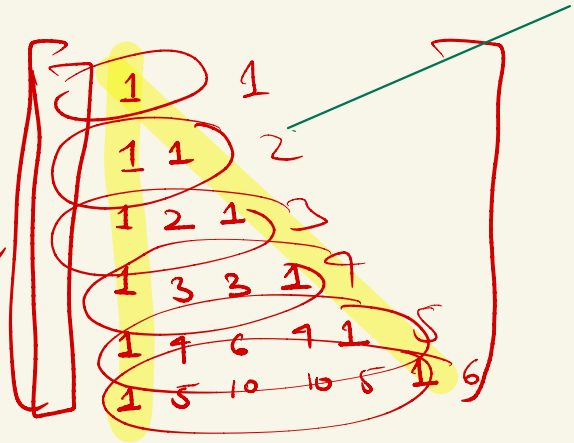
arraylist [1, 2, 3, 7, 5, 6]

↓ Remove all the Composite Number from this

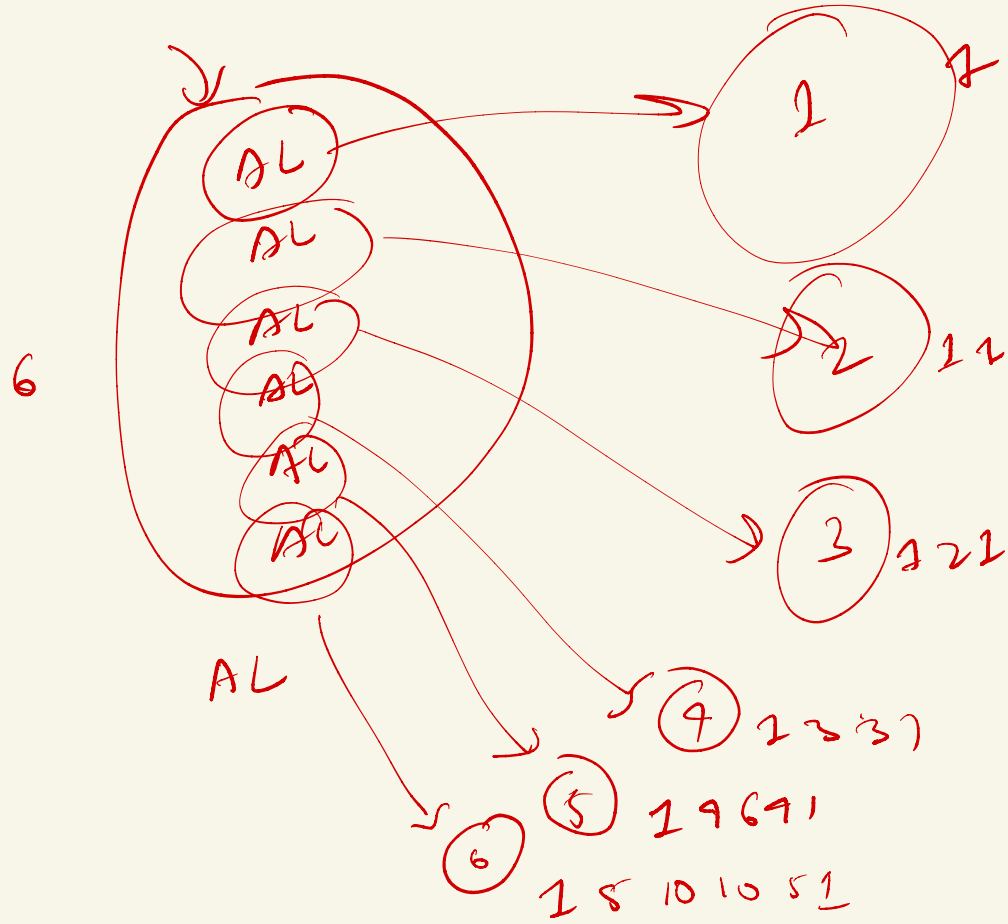
[1, 2, 3, 5]



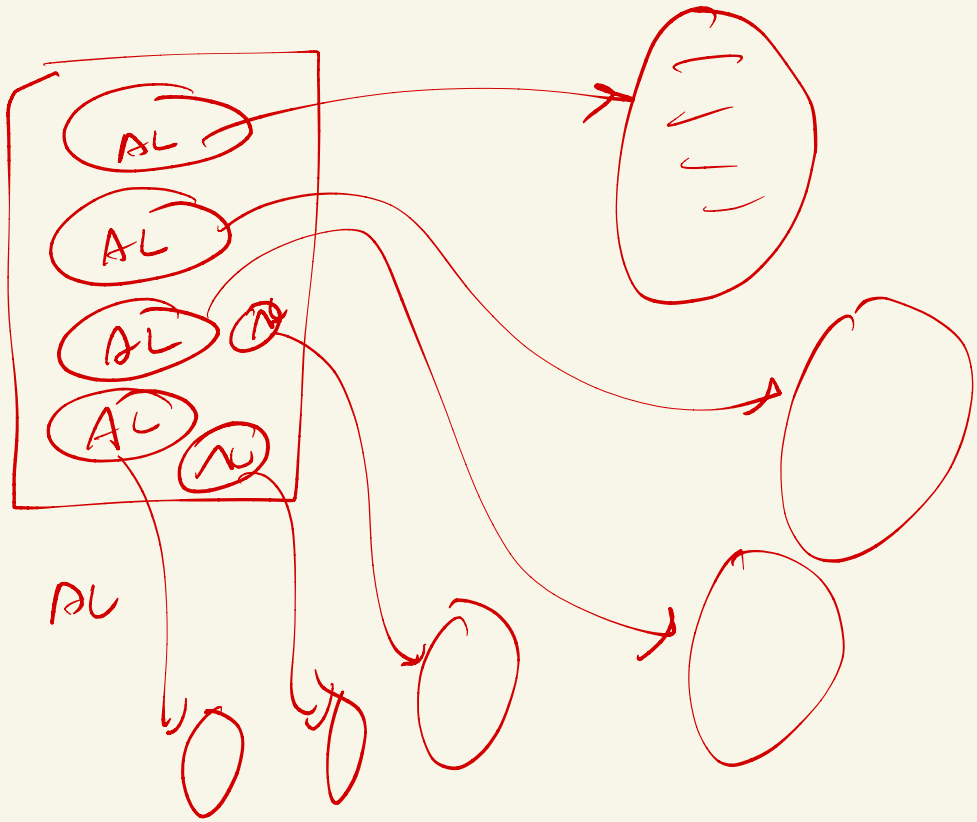
n = 6

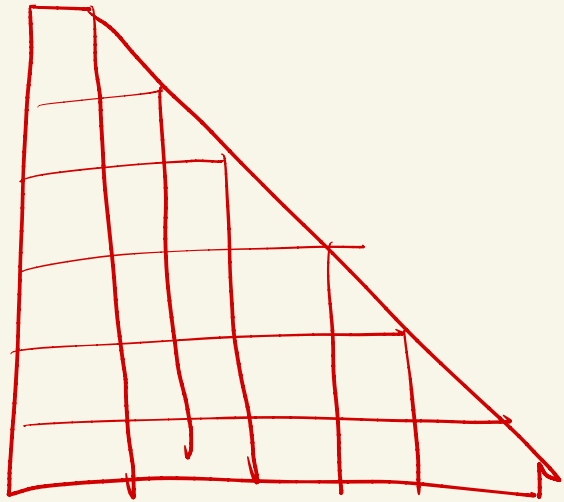

1
1
1 1
2
1 2 1
3
1 3 3 1
4
1 4 6 4 1
5
1 5 10 10 5 1
6


n
ArrayList
ArrayList<Integer>

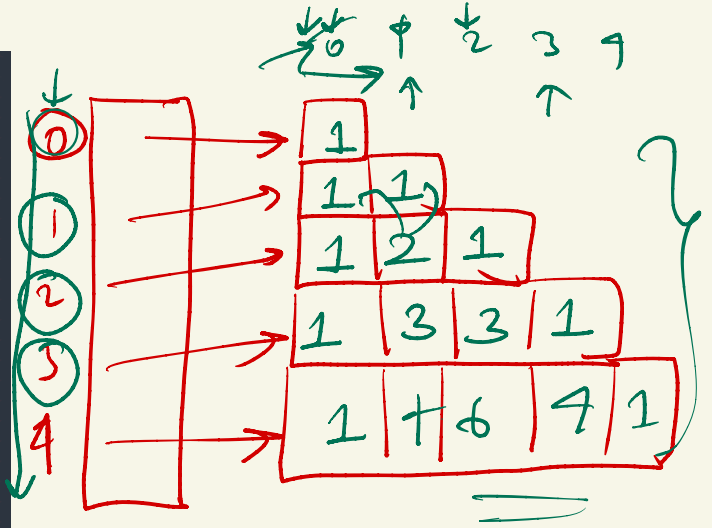

6
AL
AL
AL
AL
AL
AL
1
2
1 1
3
1 2 1
4
1 3 3 1
5
1 4 6 4 1
6
1 5 10 10 5 1

6



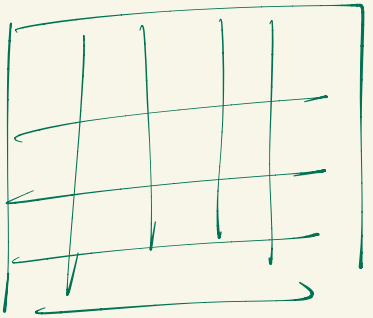


```java
public static ArrayList<ArrayList<Integer>> pascalTriangle(int n) {
    // write code here
    ArrayList<ArrayList<Integer>> pT = new ArrayList<>(n);
    for (int i = 0; i < n; i++) {
        pT.add(new ArrayList<Integer>(i + 1));
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j <= i; j++) {
            if (j == 0 || j == i) {
                pT.get(i).add(1);
            } else {
                int getAbove = pT.get(i - 1).get(j);
                int aboveDiag = pT.get(i - 1).get(j - 1);
                pT.get(i).add(getAbove + aboveDiag);
            }
        }
    }

    return pT;
}
```

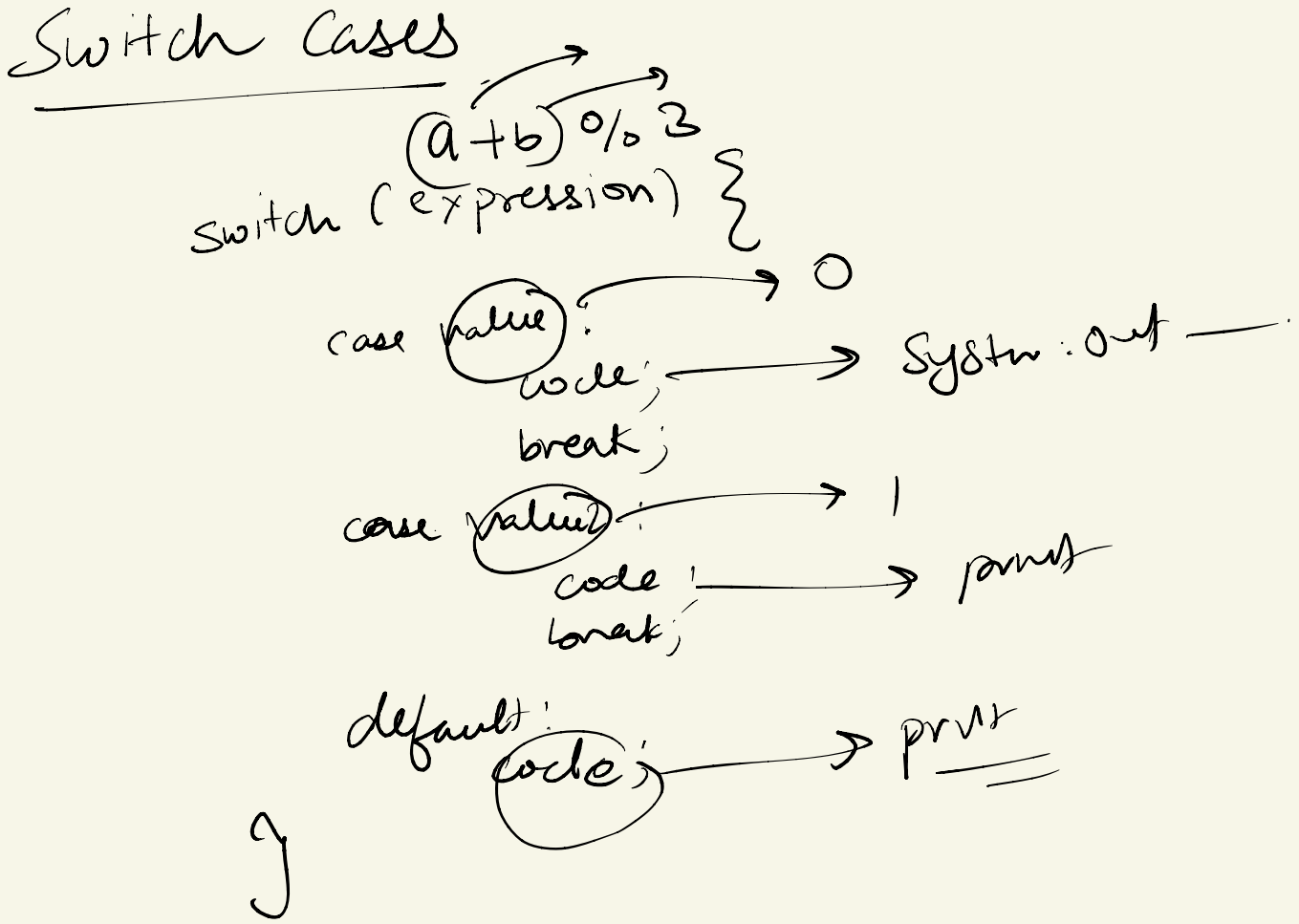


pT

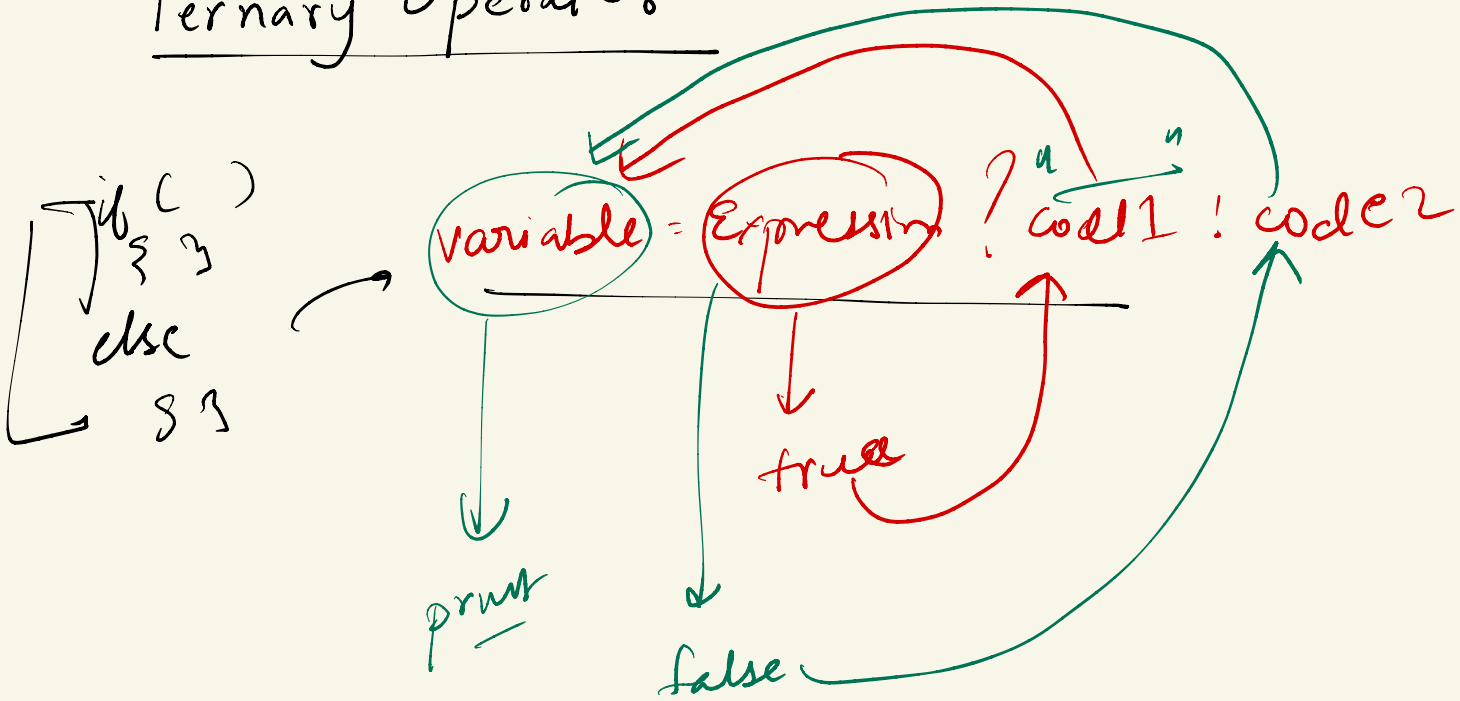
2D ArrayList

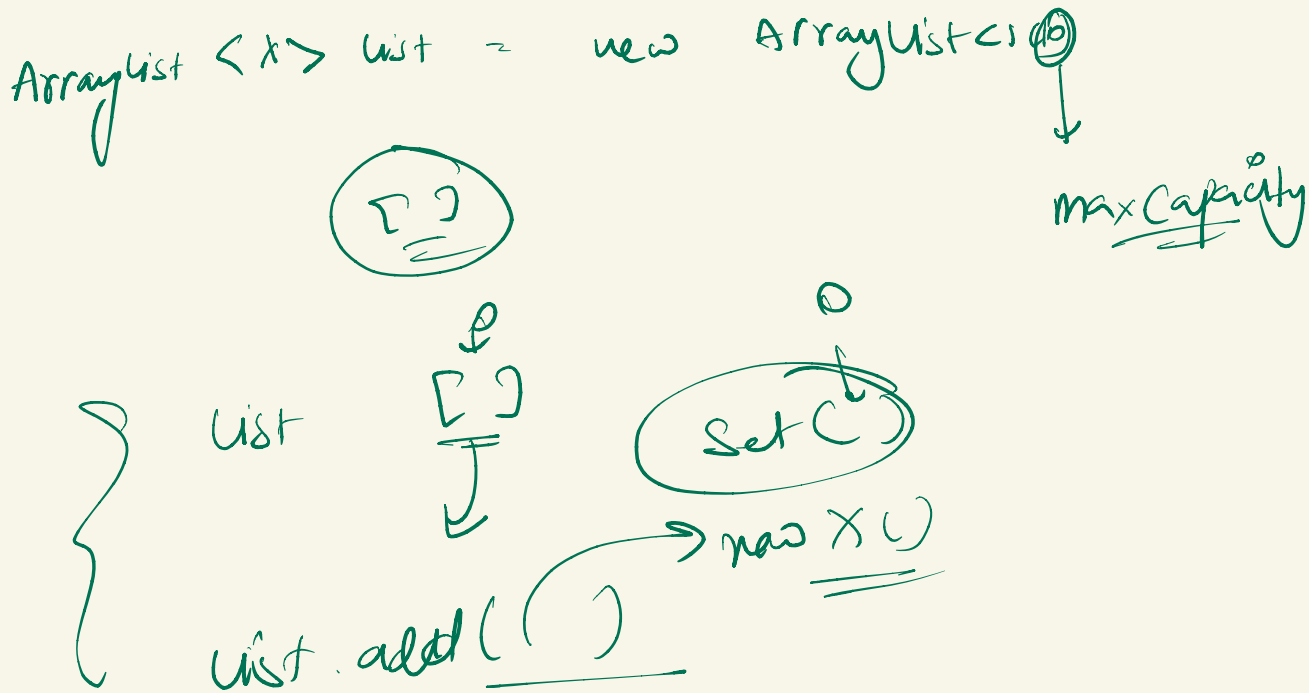


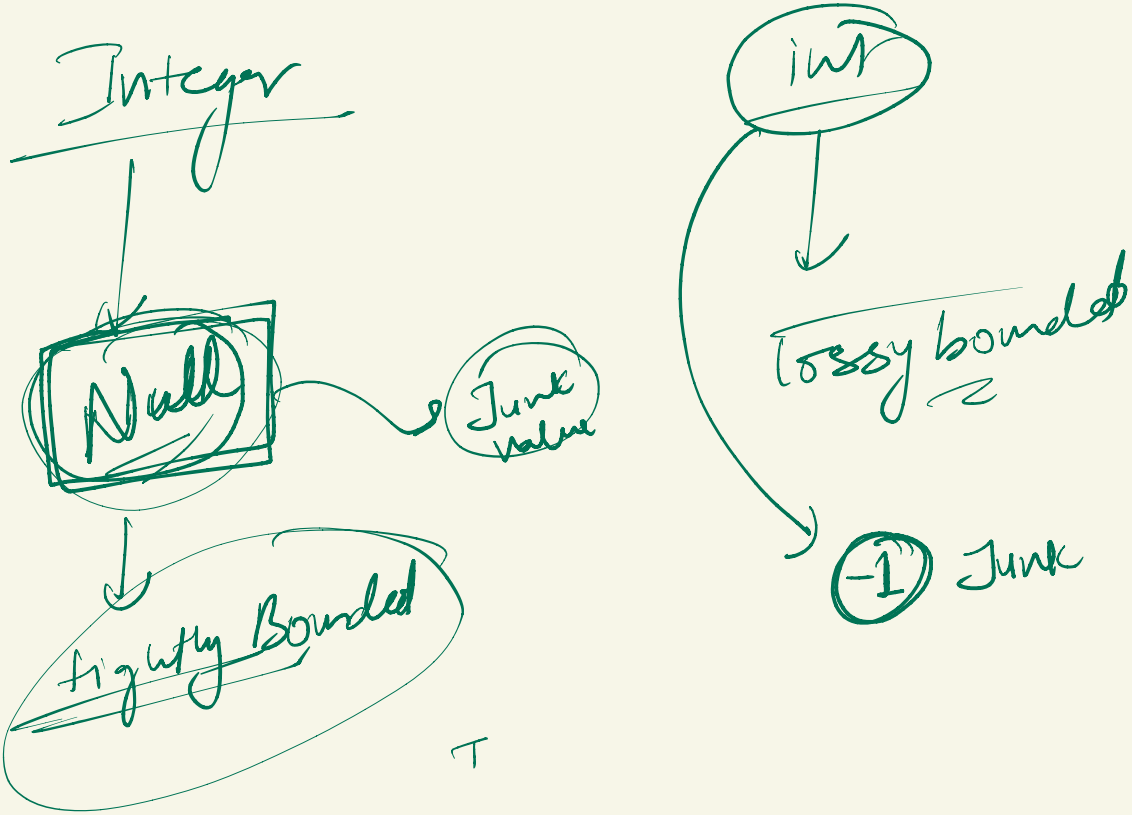
Switch cases

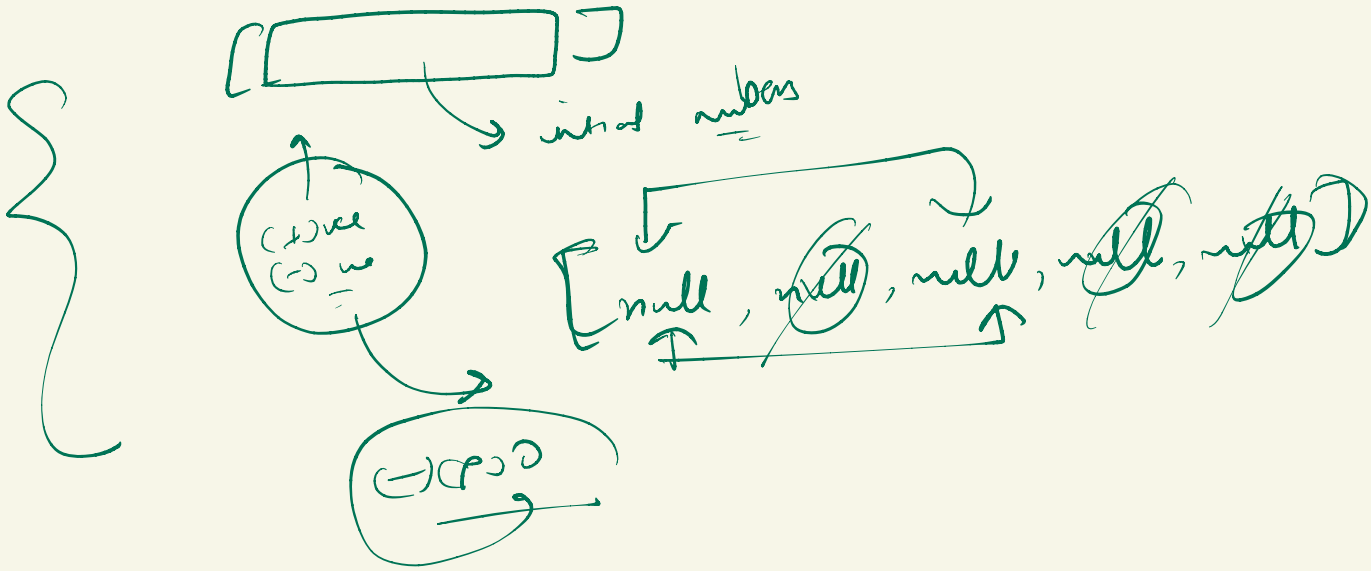


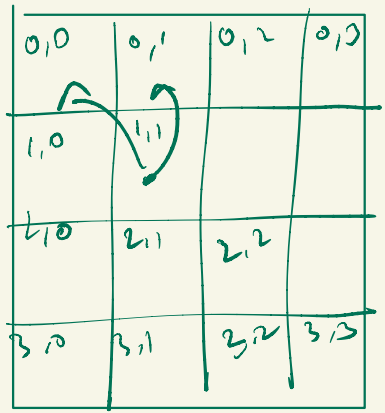
Ternary Operator




ArrayList<X> list = new ArrayList<>(10)
maxCapacity
[ ]
e
[ ]
list
0
set( )
new X( )
list.add( )

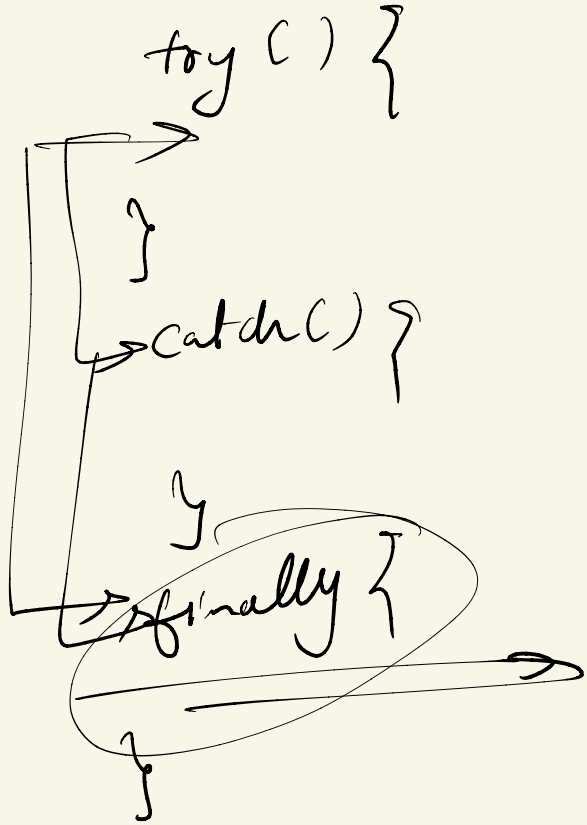

Integer
Null
Junk value
Tightly Bounded
T
int
Lossy bounded
-1
Junk





(i-1, j)

(i-1, j-1)


try {
}
catch() {
}
finally {
}