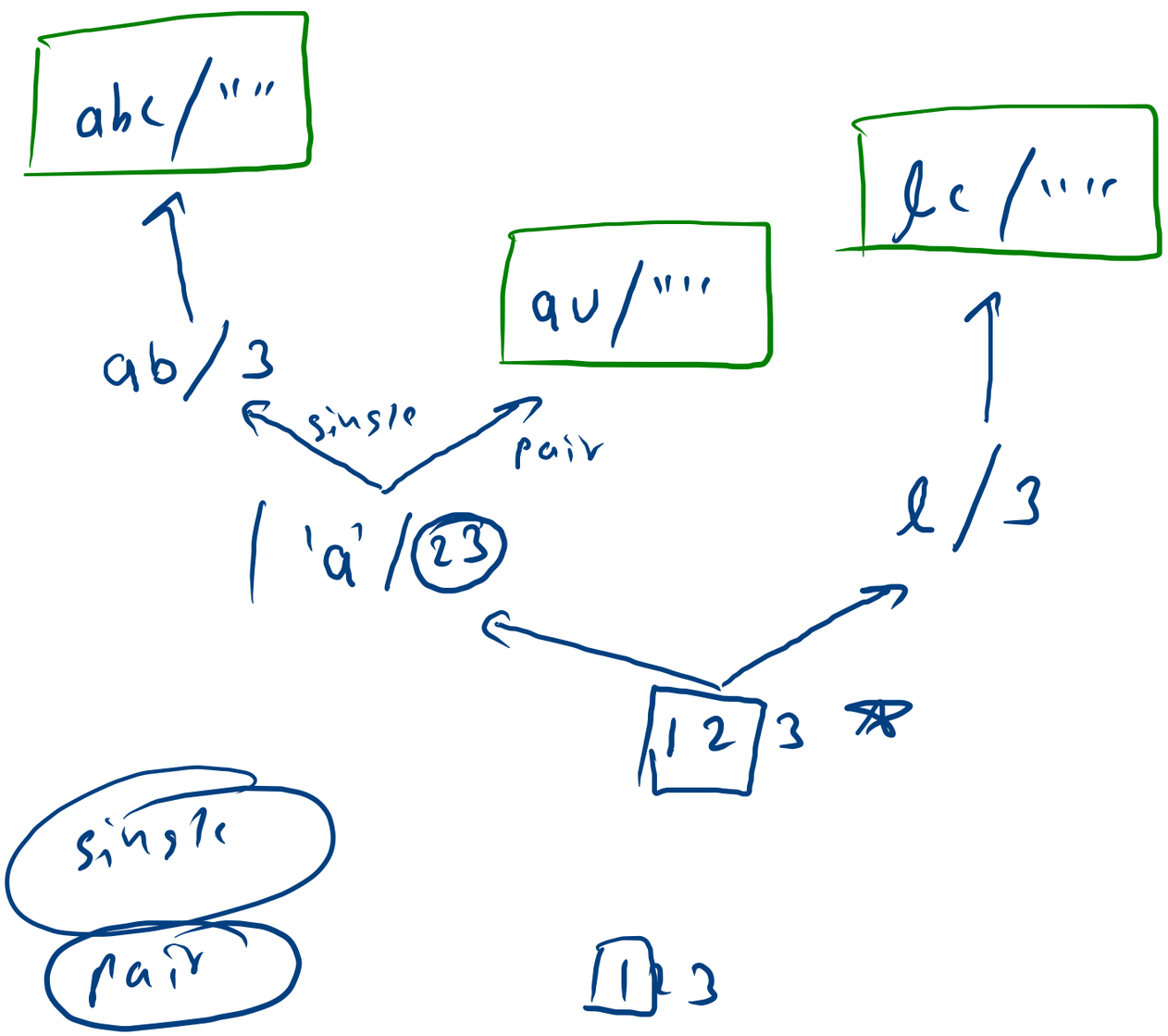
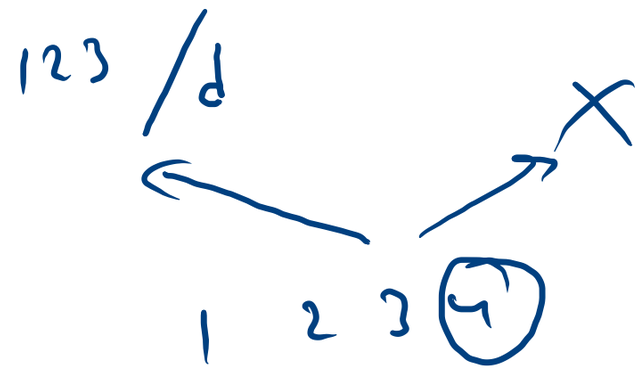
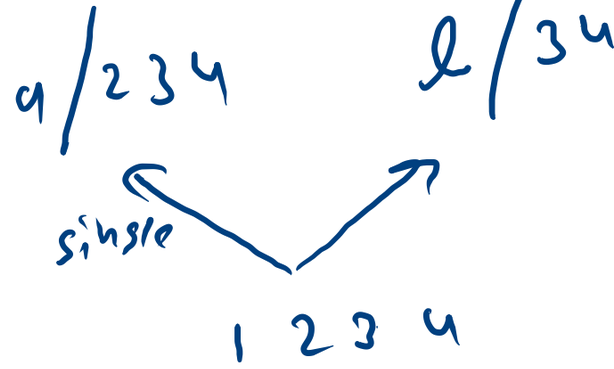
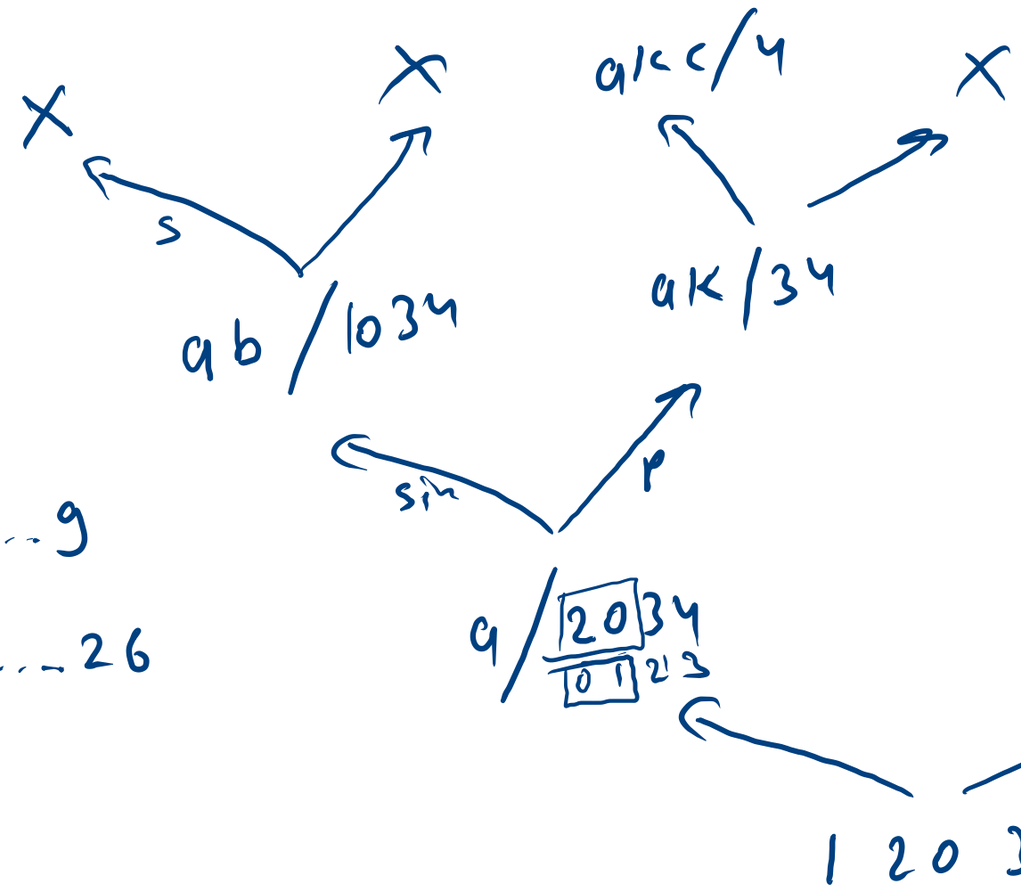
abc

q — 1
b — 2
c — 3
d — 4
e — 5
f 6
g 7
h 8
i 9
j 10
k 11
l 12
v 23
z 26









single 1...9

pair 10...26

```
int n = charAt(0) - '0';
String s = str.substring(0, 2);
int n2 = Integer.parseInt(s);
```

1 2

1 → 'a'

2 → 'b'

3 → 'c'

4 → 'd'

(0 + 'a')

(1 + 'a')

(2 + 'a')

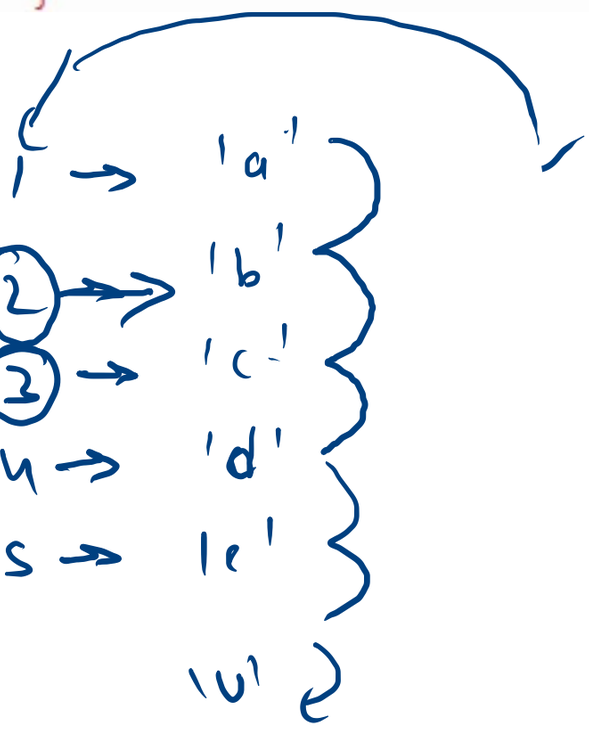
(3 + 'a')

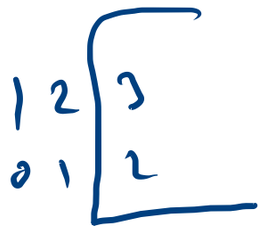
val

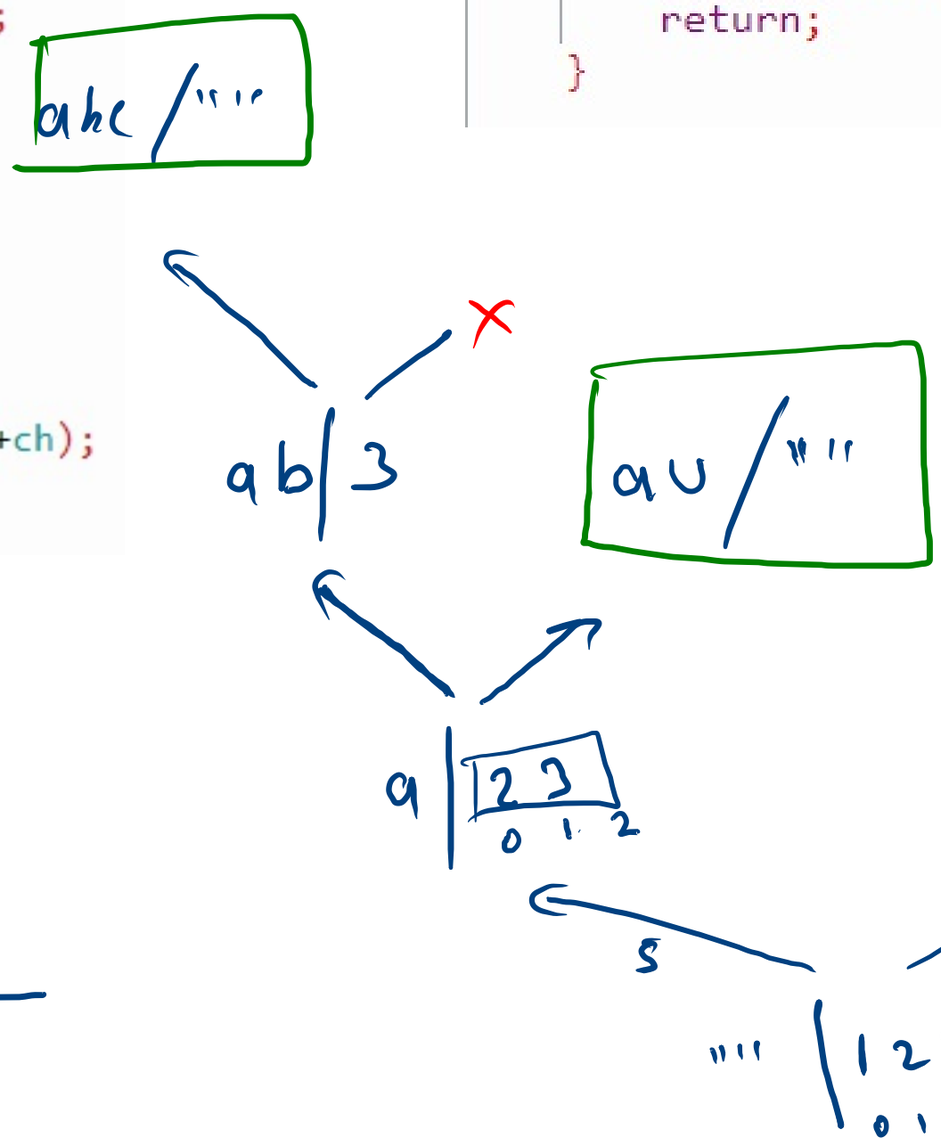
char ch = (char)(val - 1 + 'a')

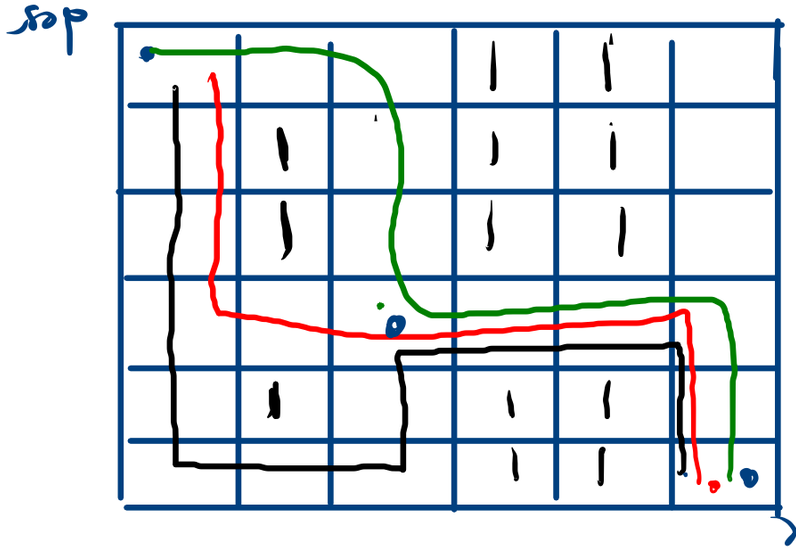
```java
int x = Integer.parseInt(str.charAt(0)+"");
if(x>=1 && x<= 9){
    char ch  = (char)(x-1+'a');
    printEncodings(str.substring(1), asf+ch);
}

// pair
if(str.length() >= 2){
    String s = str.substring(0,2);
    int val = Integer.parseInt(s);
    if(val >= 10 && val <= 26){
        char ch  = (char)(val-1+'a');
        printEncodings(str.substring(2), asf+ch);
    }
}
```

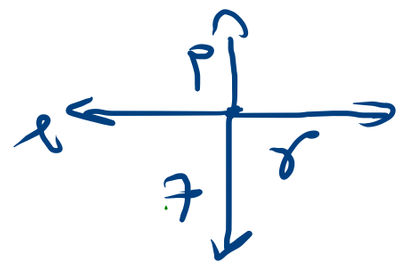
```java
if(str.length() == 0){
    System.out.println(asf);
    return;
}
```



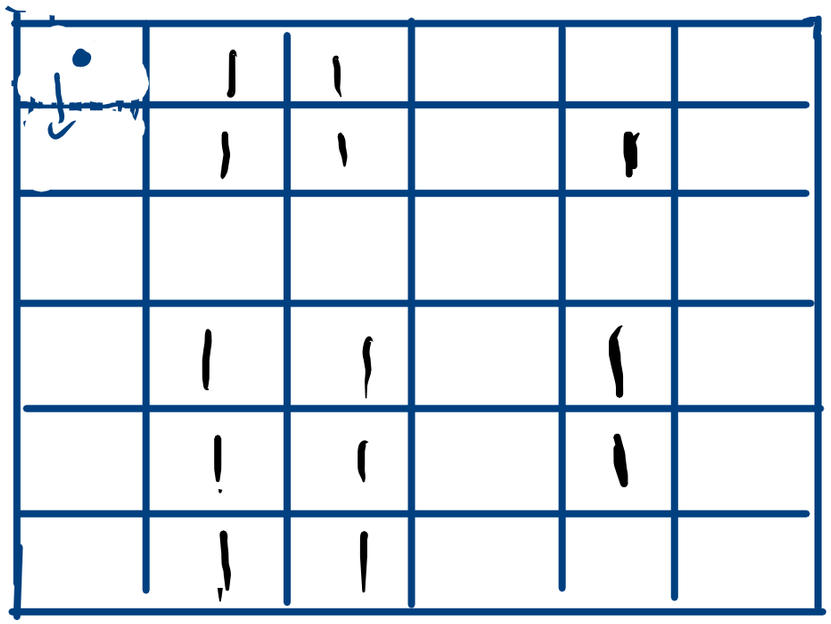


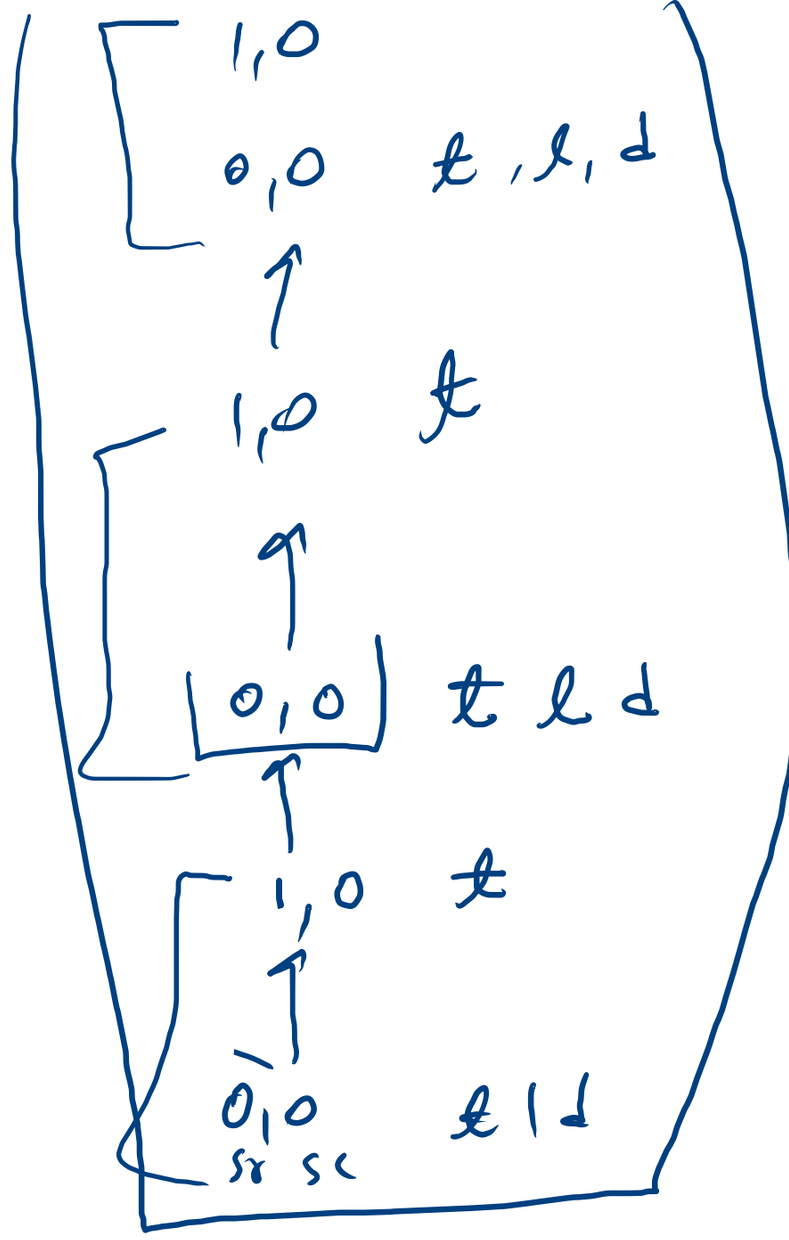




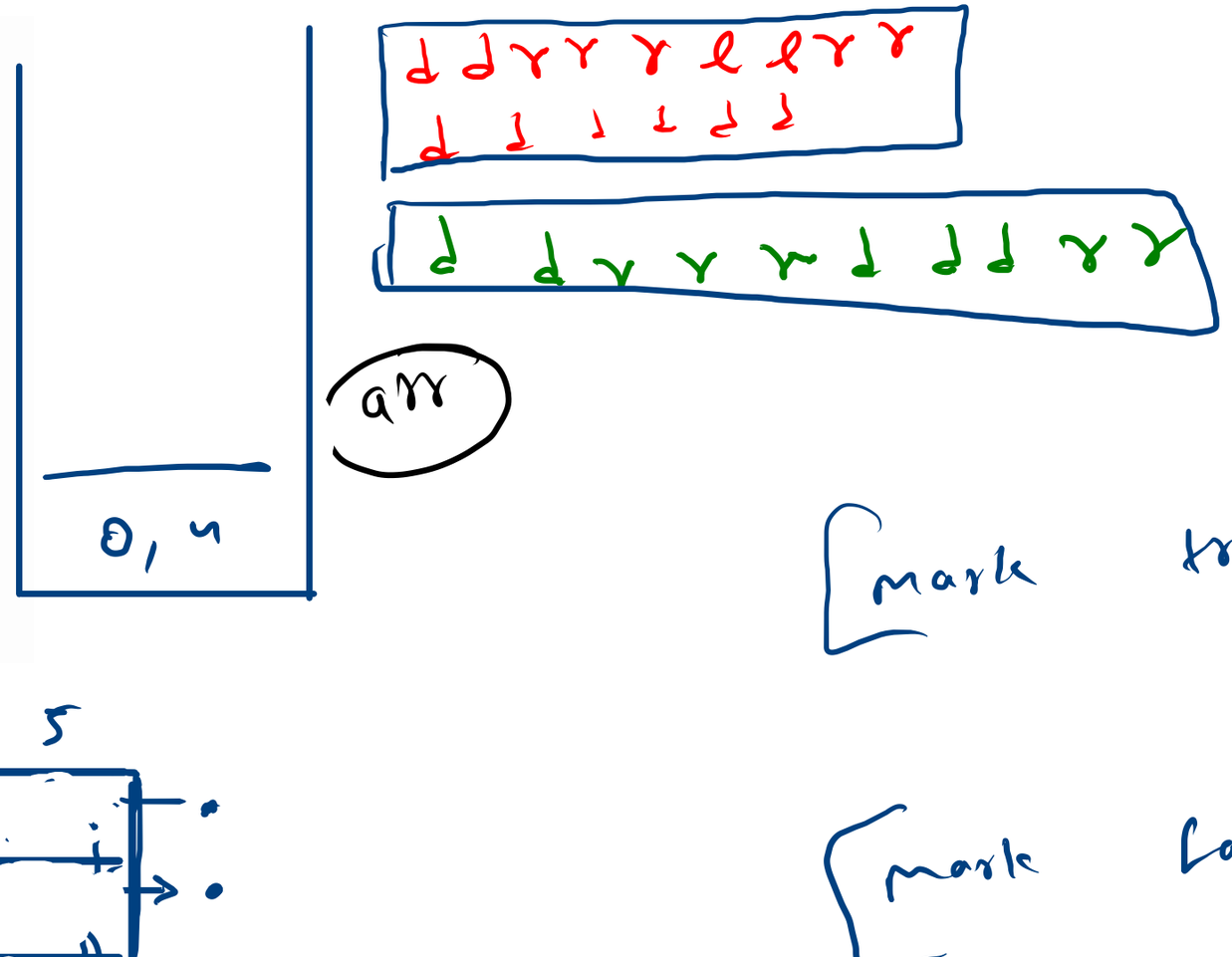


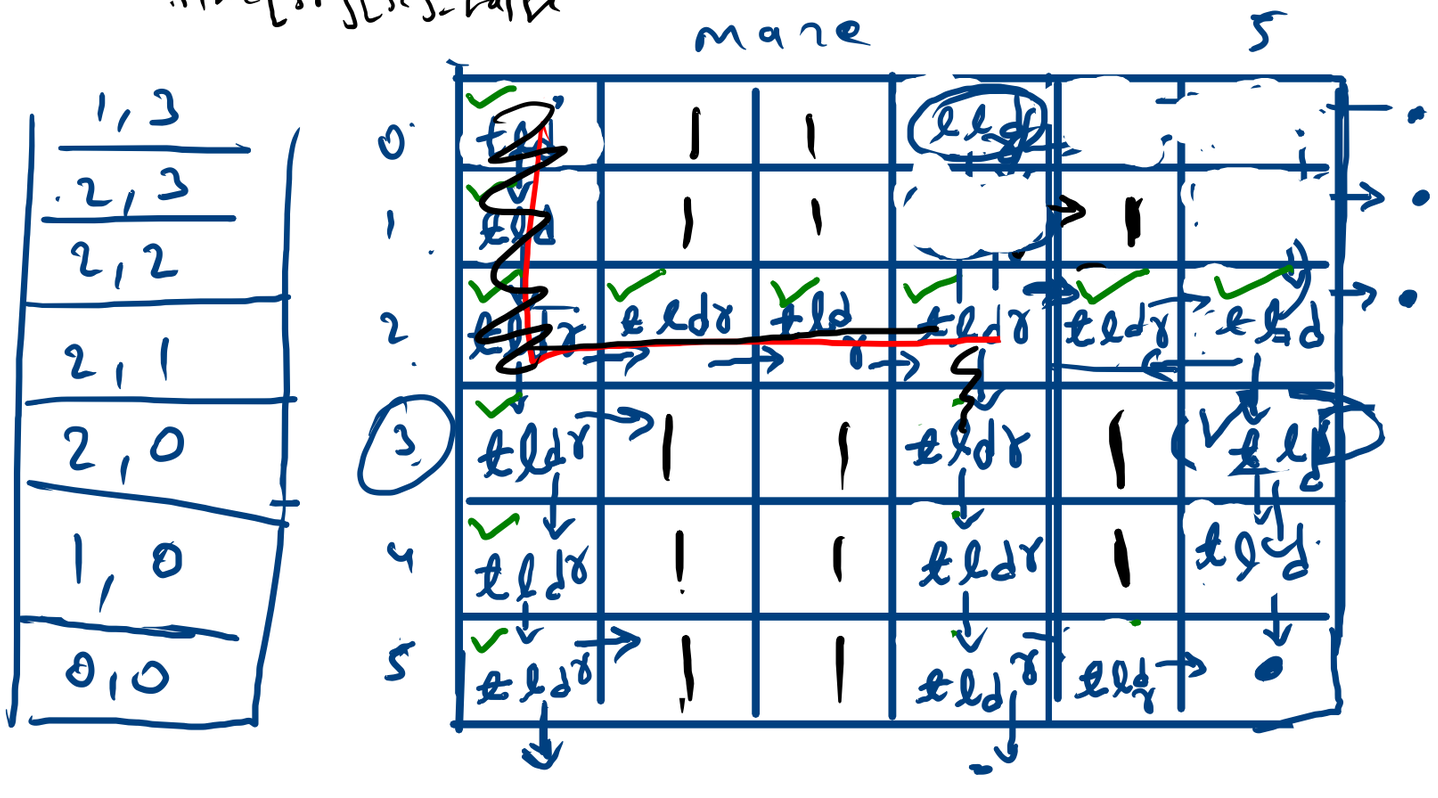
```java
public static void floodfill(int[][] maze, int sr, int sc, String asf) {
    if(sr<0||sc<0 || sr>= maze.length || sc >= maze[0].length || maze[sr][sc]==1)return;
       1     0      1         6           0         6
    if(sr==maze.length-1 && sc == maze[0].length-1){
        System.out.println(asf);
        return;
    }

    floodfill(maze, sr-1, sc, asf+"t");// t
    floodfill(maze, sr, sc-1, asf+"l");// l
    floodfill(maze, sr+1, sc, asf+"d");// d
    floodfill(maze, sr, sc+1, asf+"r");// r
}
```



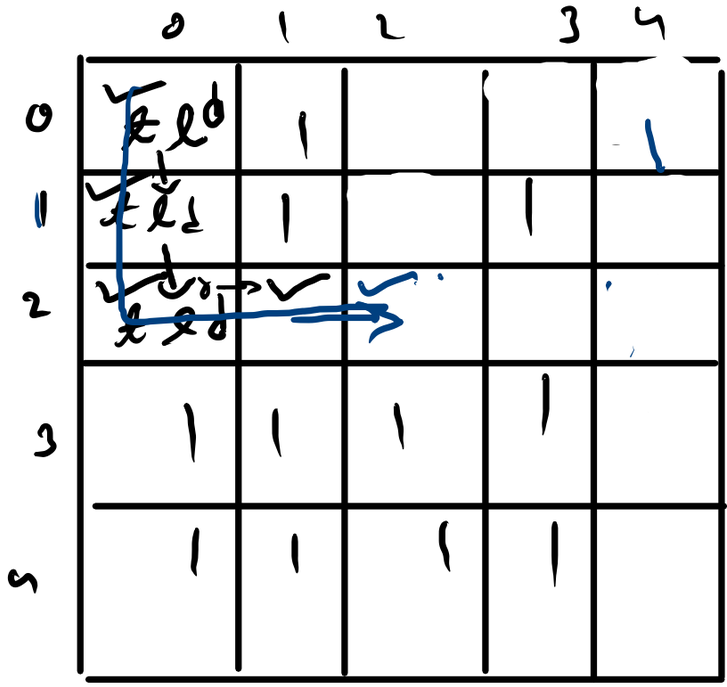


```java
if(sr<0||sc<0 || sr>= maze.length ||
sc >= maze[0].length || maze[sr][sc]==1 ||
visited[sr][sc] == true)return;

if(sr==maze.length-1 && sc == maze[0].length-1){
    System.out.println(asf);
    return;
}

visited[sr][sc] = true;
floodfill(maze, sr-1, sc, asf+"t", visited);// t
floodfill(maze, sr, sc-1, asf+"l", visited);// l
floodfill(maze, sr+1, sc, asf+"d", visited);// d
floodfill(maze, sr, sc+1, asf+"r", visited);// r
```





```java
if(sr<0||sc<0 || sr>= maze.length ||
sc >= maze[0].length || maze[sr][sc]==1 ||
visited[sr][sc] == true)return;

if(sr==maze.length-1 && sc == maze[0].length-1){
    System.out.println(asf);
    return;
}

visited[sr][sc] = true;
floodfill(maze, sr-1, sc, asf+"t", visited);// t
floodfill(maze, sr, sc-1, asf+"l", visited);// l
floodfill(maze, sr+1, sc, asf+"d", visited);// d
floodfill(maze, sr, sc+1, asf+"r", visited);// r
visited[sr][sc] = false;
```



2,2	"drr"	tldrr
2,1	"dr"	tldr
2,0	"d"	tld
1,0	"d"	tld
0,0	""	tld

asf

0 ... 20

+V ←

0 ←

[10 20 30 40 50]

60

target = 60 ←

[10 20 30 40 50]

↓

d

shell

10, 20, 30, .

10, 50, .

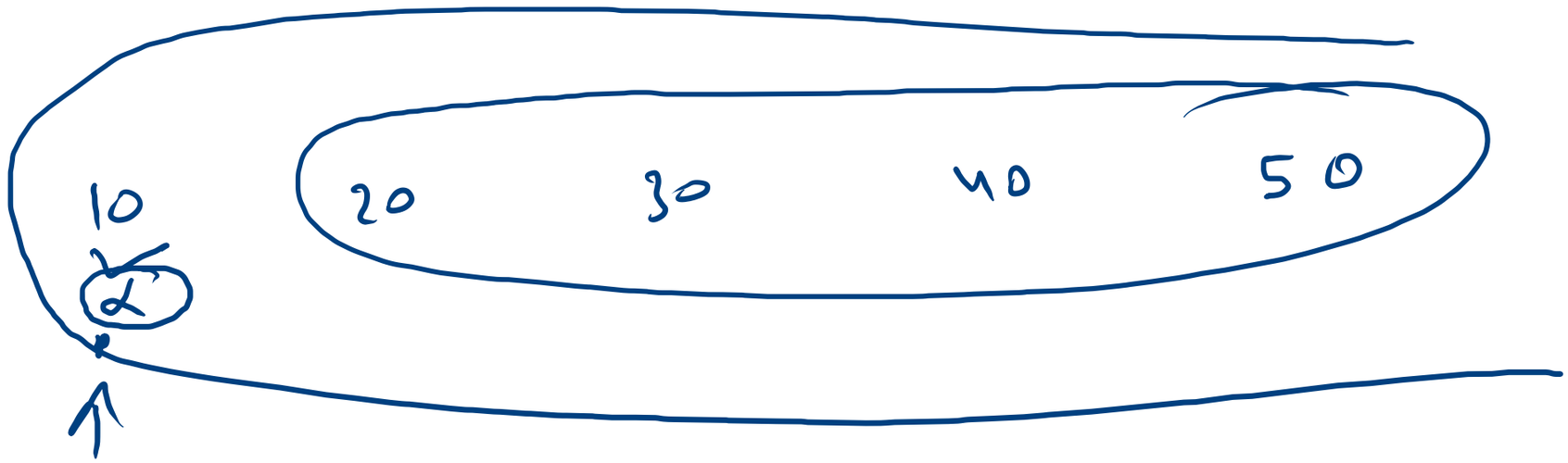
20, 40, .

print

we only once

10 → 20,30,50

— , 20 , 40



50

60

expression ind = 0

tar → 60

fair

ind+1

tar = 50

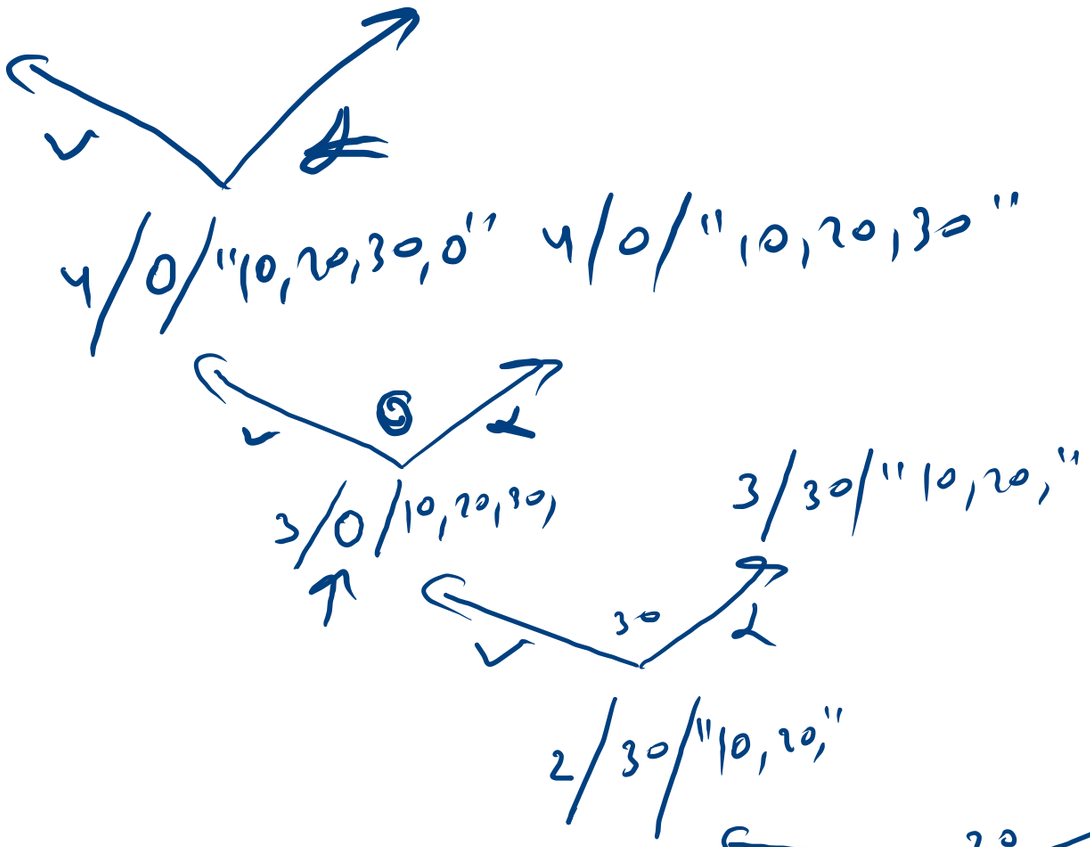
ind+1

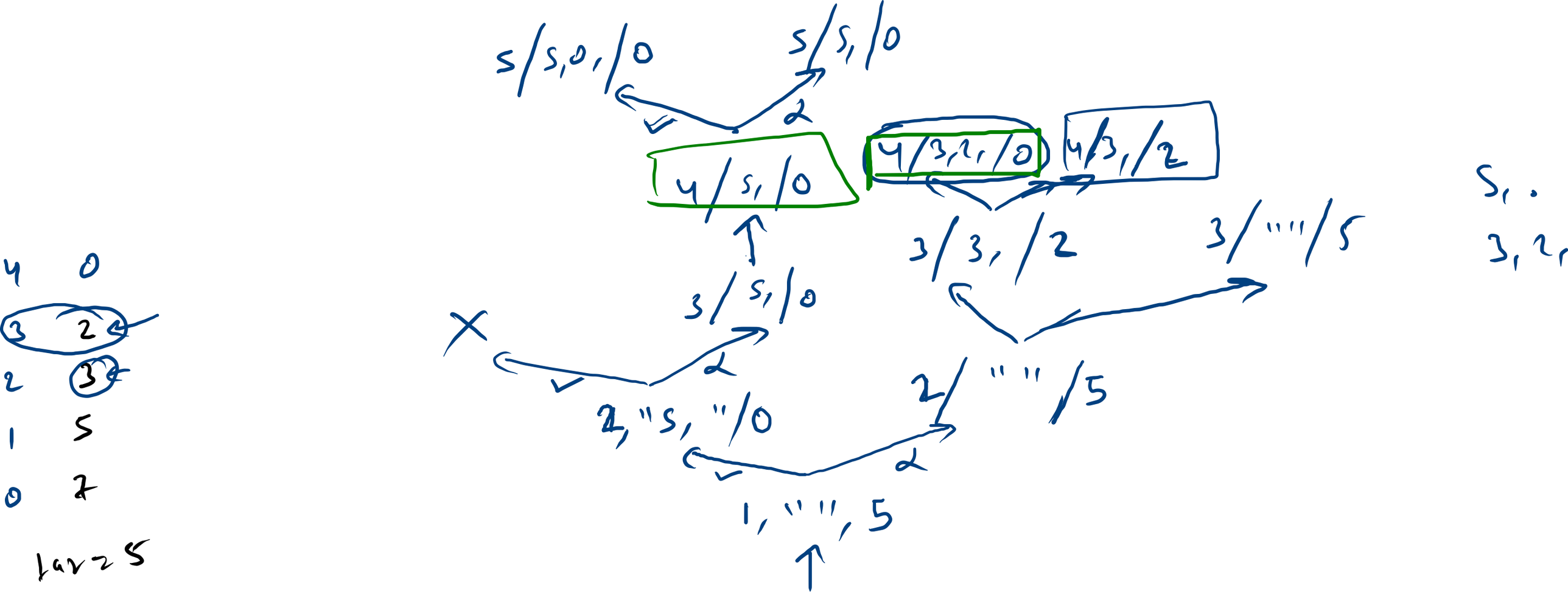
tar = 60

6

5

subset


4/0/"10,20,30,0"
4/0/"10,20,30"
3/0/10,20,30,
3/30/"10,20,"
2/30/"10,20,"
2/50/"10,"
1/50/"10,"
1/60/""
0/60/""



4 0

3 2

2 3

1 5

0 7

1av25

```
public static void printTargetSumSubsets(int[] arr, int idx, String set, int tar) {
    if(idx == arr.length){
        if(tar == 0)
            System.out.println(set+".");
        return;
    }
    // include
    if(arr[idx] <= tar)
        printTargetSumSubsets(arr, idx+1, set+arr[idx]+", ", tar-arr[idx]);
    // xclude
    printTargetSumSubsets(arr, idx+1, set, tar);

}
```