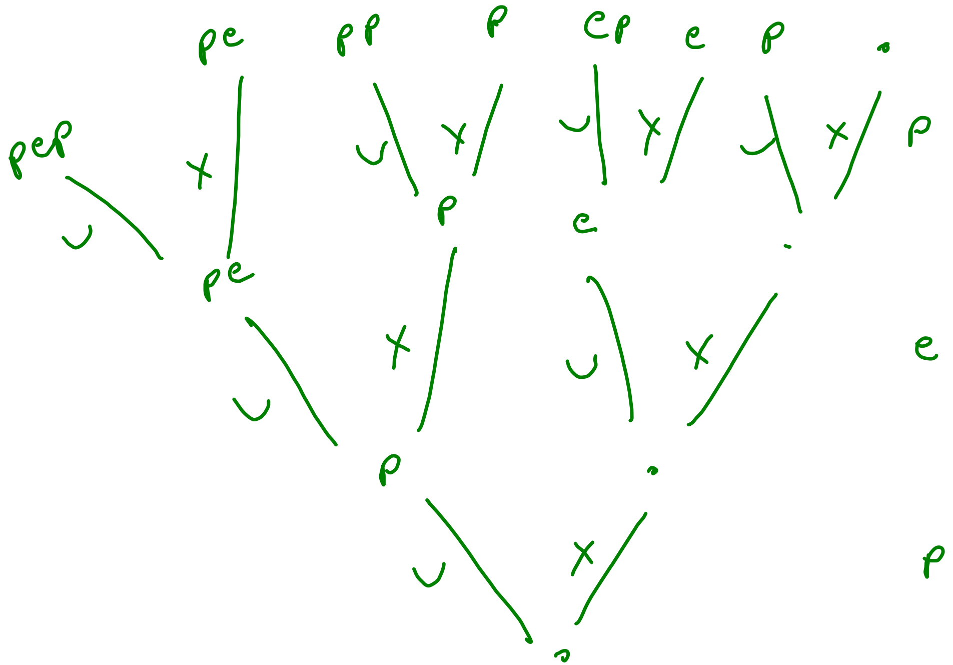
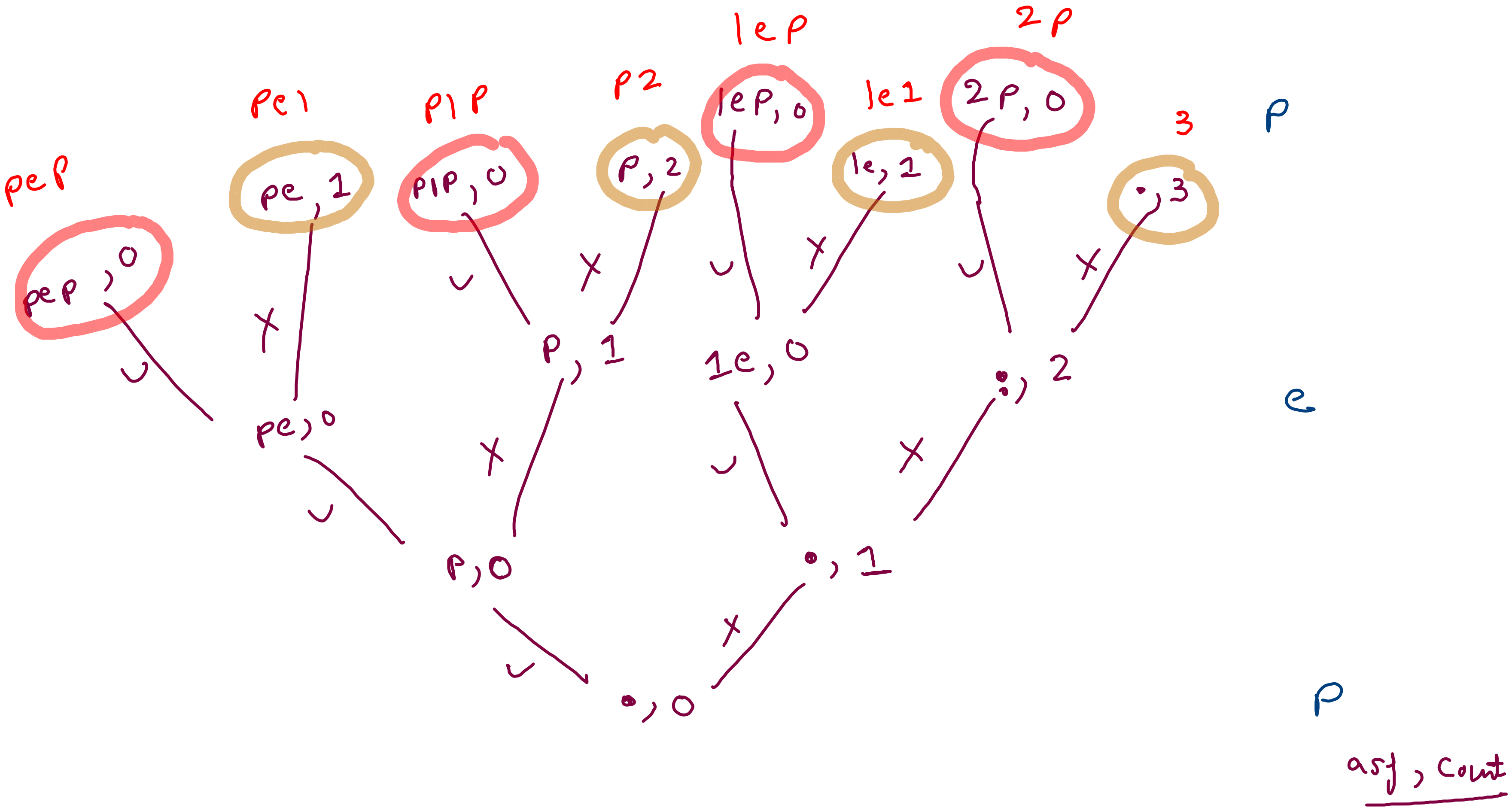


pep

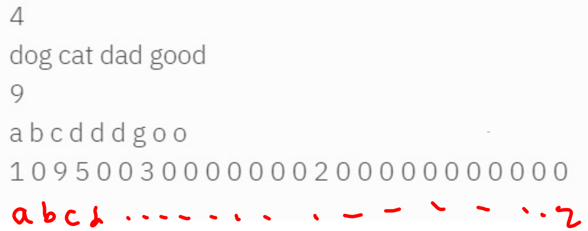
000	→	p e p
001	→	p e 1
010	→	p 1 p
011	→	p 2
100	→	1 e p
101	→	1 e 1
110	→	2p
111	→	3

1 → replace

0 → no change



pep
pep, 0
pe1
pe, 1
p1p
p1p, 0
p2
p, 2
lep
1ep, 0
le1
1e, 1
2p
2p, 0
3
., 3
pe, 0
p, 1
1e, 0
., 2
p, 0
., 1
., 0
p
e
p
asf, count



```
4
dog cat dad good
9
a b c d d d g o o
1 0 9 5 0 0 3 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0
```

a b c d . . . . z

a -> 2

d -> 4

dog cat dad good X (a > 2)

dog — dad good X (d > 3)

dog — — good X (o > 3)

— — dad good ✓

dog — dad — ✓

2^{16}

↓ valid

dad good

d -> 3

a -> 1

g -> 1

o -> 2

15 + 1 + 3 + 4

= $\boxed{23}$

d -> 3

a -> 1

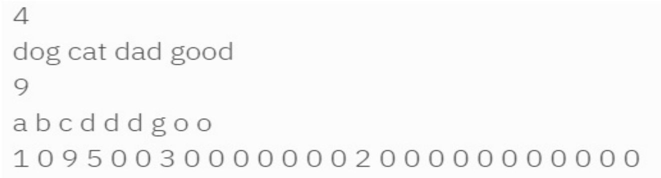
o -> 1

g -> 1

15 + 1 + 2

+ 3

= $\boxed{21}$



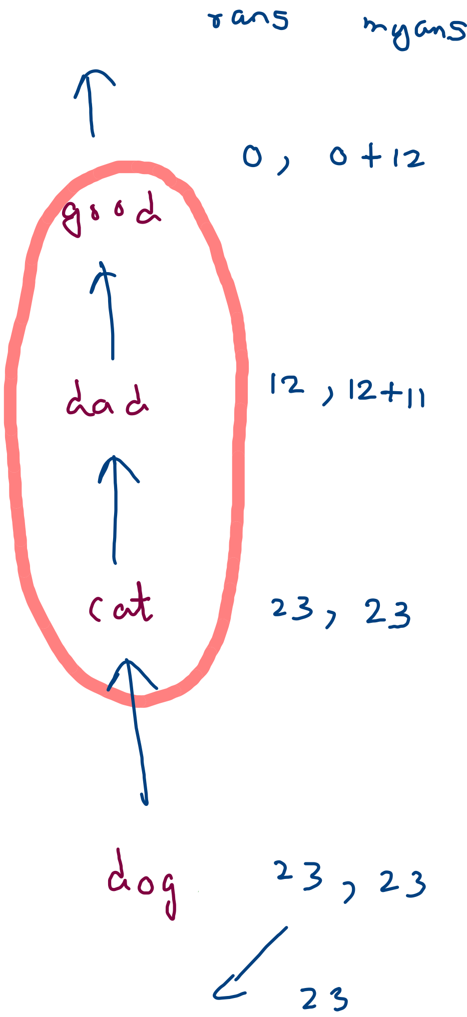
```
4
dog cat dad good
9
a b c d d d g o o
1 0 9 5 0 0 3 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0
```

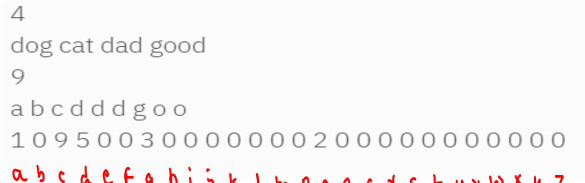
a b c d e f g h i j k l m n o p q r s t u v w x y z

1	1	1	3	0	0	1		....
0	1	2	3	4	5	6	7	8 25
↓	↓	↓	↓	↓				↓
a	b	c	d	e	- - -	- -	- -	. . . z

rans

rans, rans + score





```
4
dog cat dad good
9
a b c d d d g o o
1 0 9 5 0 0 3 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0
```

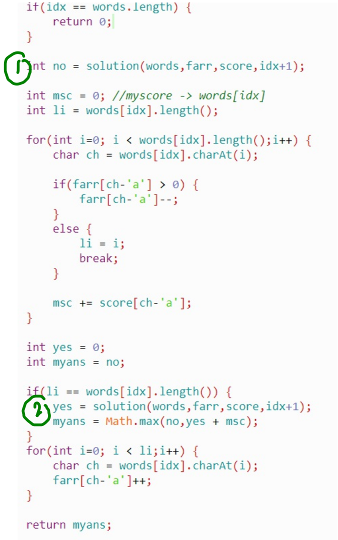
a b c d e f g h i j k l m n o p q r s t u v w x y z

a	b	c	d	e	j	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0 ~~1~~	1	1	1 ~~2~~ ~~3~~			1								2											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

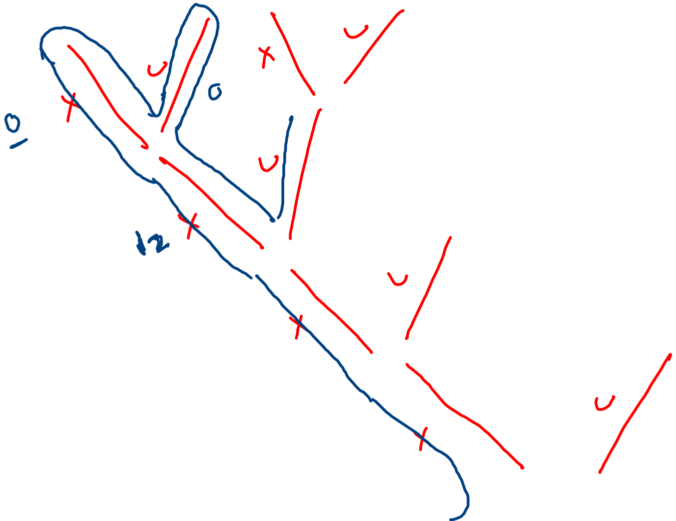
25

msc = 5 + 1 + 5

li = 3



```
if(idx == words.length) {
    return 0;
}

① int no = solution(words,farr,score,idx+1);

int msc = 0; //myscore -> words[idx]
int li = words[idx].length();

for(int i=0; i < words[idx].length();i++) {
    char ch = words[idx].charAt(i);

    if(farr[ch-'a'] > 0) {
        farr[ch-'a']--;
    }
    else {
        li = i;
        break;
    }

    msc += score[ch-'a'];
}

int yes = 0;
int myans = no;

if(li == words[idx].length()) {
② yes = solution(words,farr,score,idx+1);
    myans = Math.max(no,yes + msc);
}
for(int i=0; i < li;i++) {
    char ch = words[idx].charAt(i);
    farr[ch-'a']++;
}

return myans;
```



good

dad

cat

dog

n = 1000

1
10
100
1000
101
102
103
104
105
106
107
108
109

10

11

11
110
111
112
113
114
115
116
117
118
119

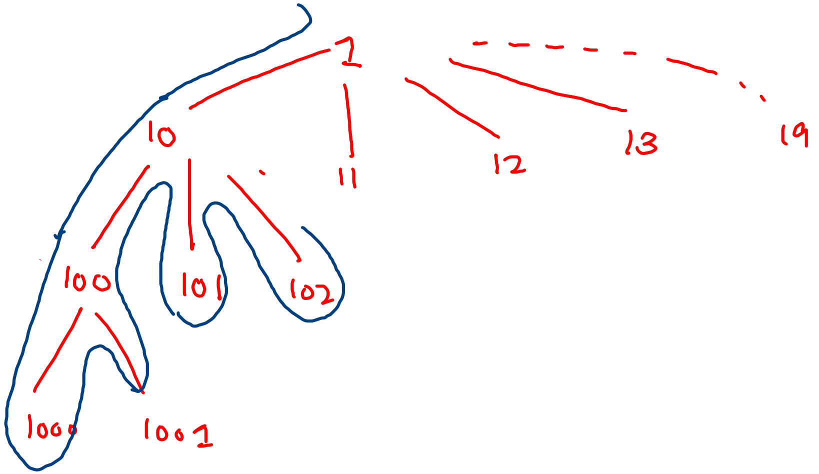
12
120
121
122
.
.
.
129

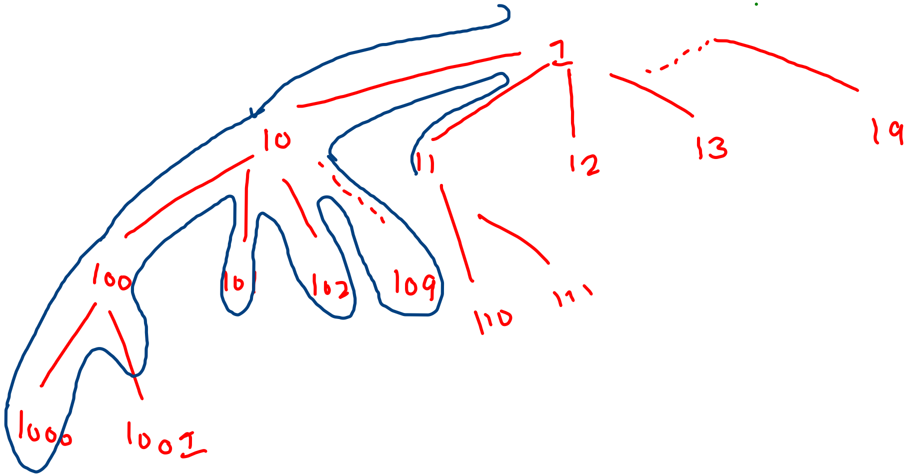
13
131

199

2
20
200
201
202
203
.
.
.
209

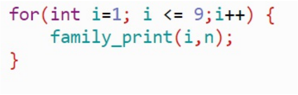
21
210



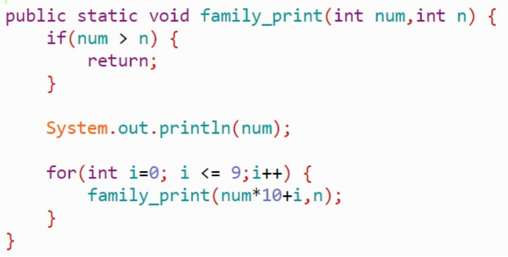


n = 1000

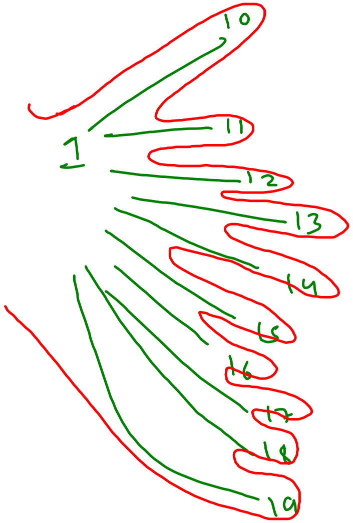
n = 25



```
for(int i=1; i <= 9;i++) {
    family_print(i,n);
}
```



```
public static void family_print(int num,int n) {
    if(num > n) {
        return;
    }

    System.out.println(num);

    for(int i=0; i <= 9;i++) {
        family_print(num*10+i,n);
    }
}
```



1

10

11

12

13

14

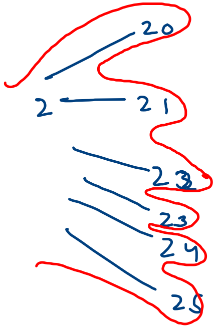
15

16

17

18

19



2

20

21

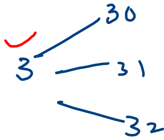
22

23

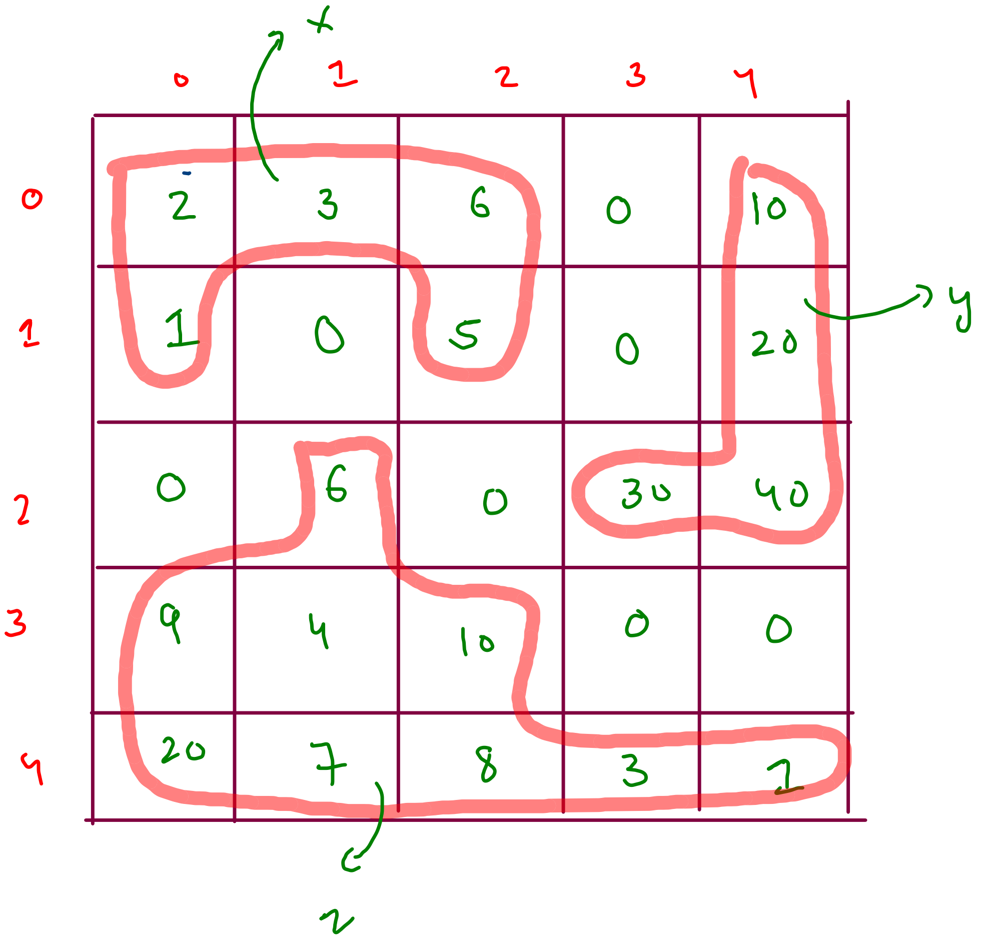
24

25

3



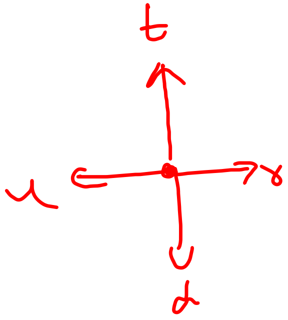
max (x, y, z)

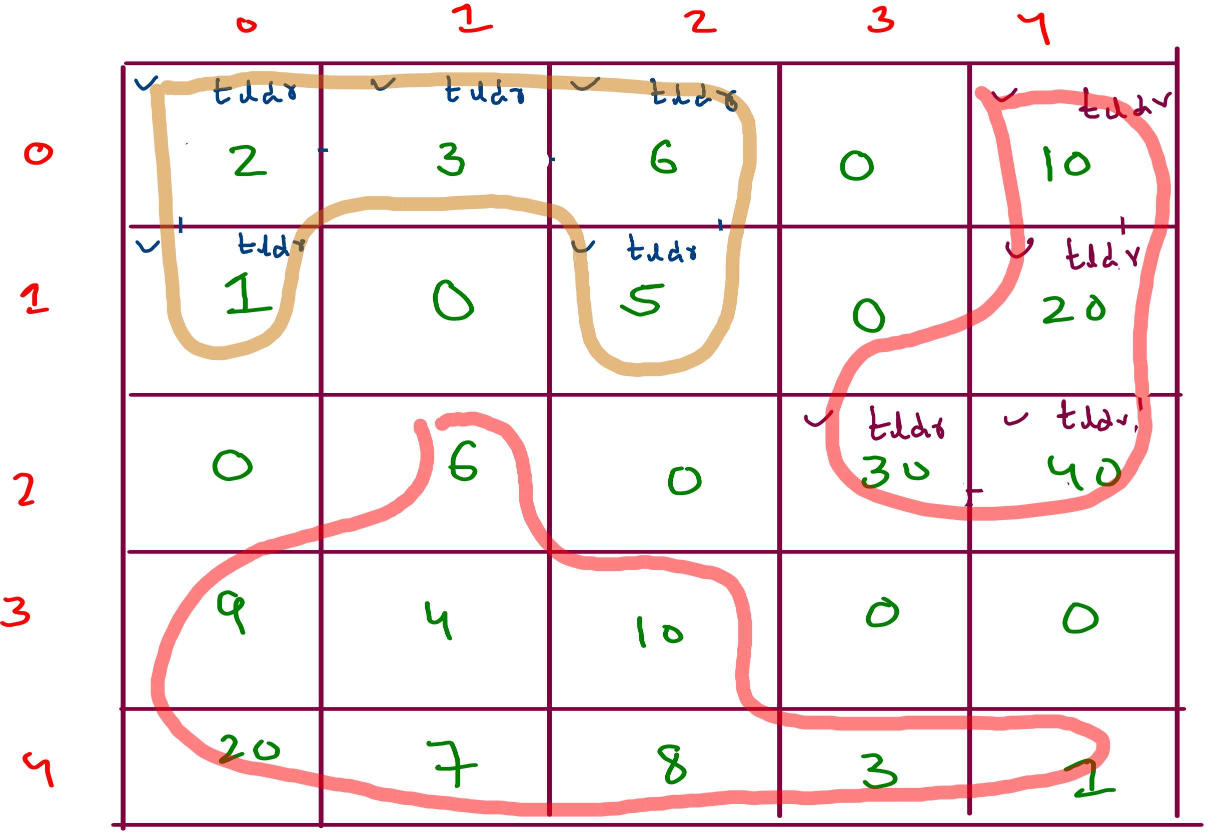


	0	1	2	3	4
0	2	3	6	0	10
1	1	0	5	0	20
2	0	6	0	30	40
3	9	4	10	0	0
4	20	7	8	3	1

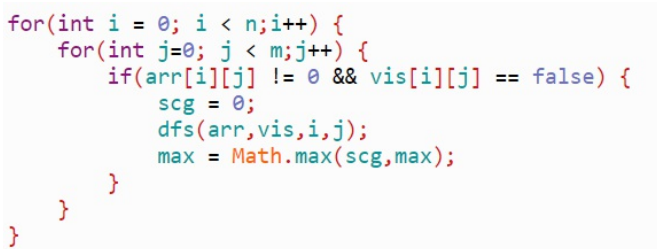
+ve -> gold

0 -> blocked

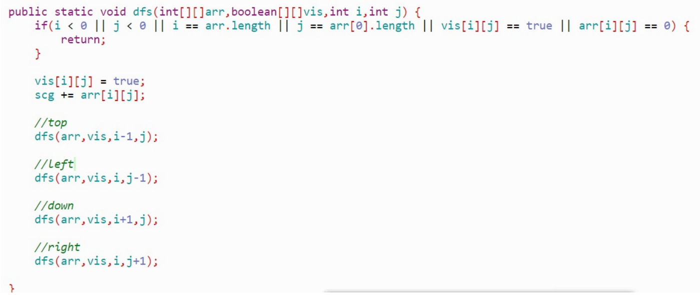




	0	1	2	3	4
0	2	3	6	0	10
1	1	0	5	0	20
2	0	6	0	30	40
3	9	4	10	0	0
4	20	7	8	3	1



```java
for(int i = 0; i < n;i++) {
    for(int j=0; j < m;j++) {
        if(arr[i][j] != 0 && vis[i][j] == false) {
            scg = 0;
            dfs(arr,vis,i,j);
            max = Math.max(scg,max);
        }
    }
}
```



```java
public static void dfs(int[][]arr,boolean[][]vis,int i,int j) {
    if(i < 0 || j < 0 || i == arr.length || j == arr[0].length || vis[i][j] == true || arr[i][j] == 0) {
        return;
    }

    vis[i][j] = true;
    scg += arr[i][j];

    //top
    dfs(arr,vis,i-1,j);

    //left
    dfs(arr,vis,i,j-1);

    //down
    dfs(arr,vis,i+1,j);

    //right
    dfs(arr,vis,i,j+1);

}
```

max = ~~17~~ 100

scg =