str → yvTA

$2^4 \rightarrow 16$

- yvTA
- yvT
- yvA
- yv
- yTA
- yT
- yA
- y

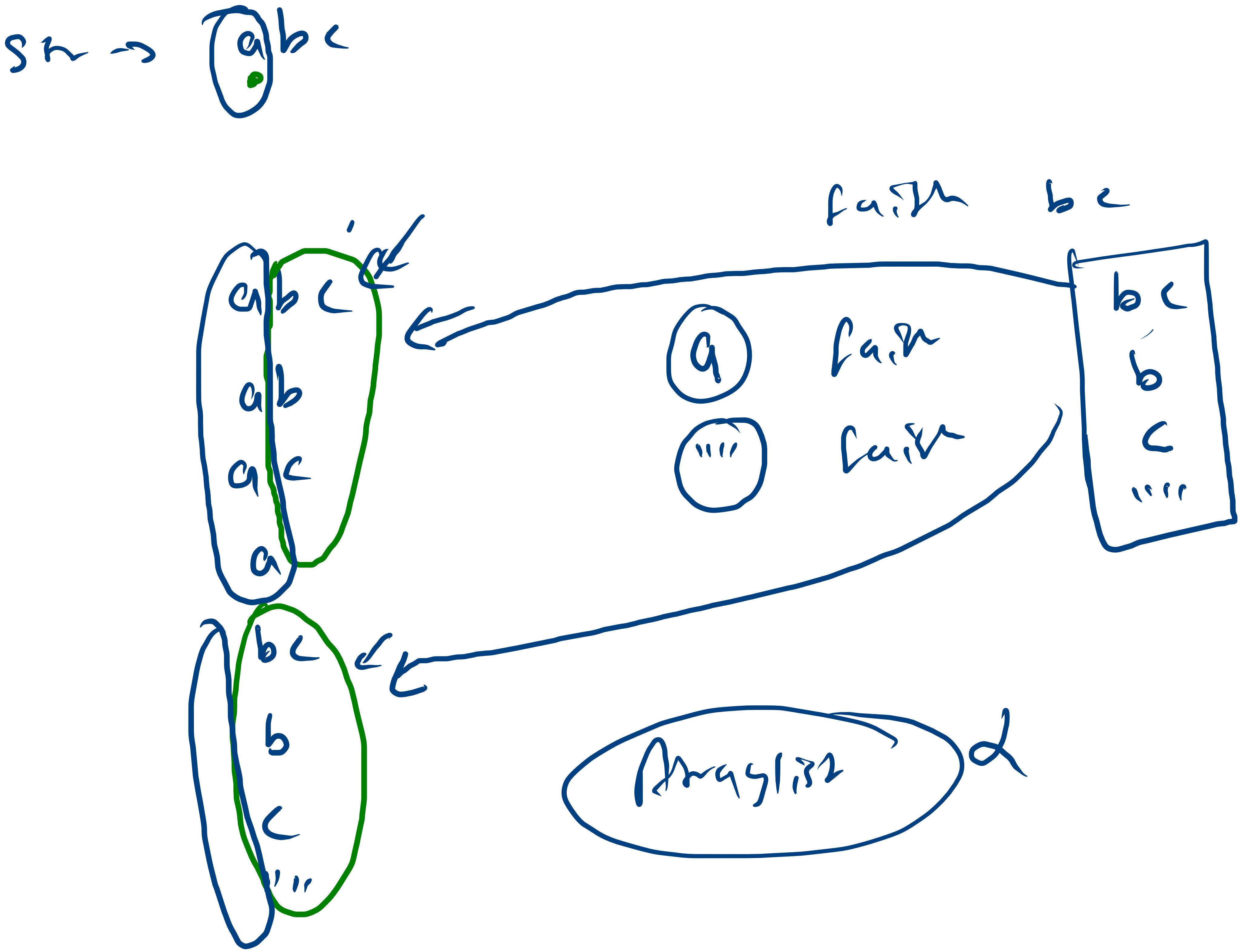
1111
1110
1101
1100
1011
1010
1001

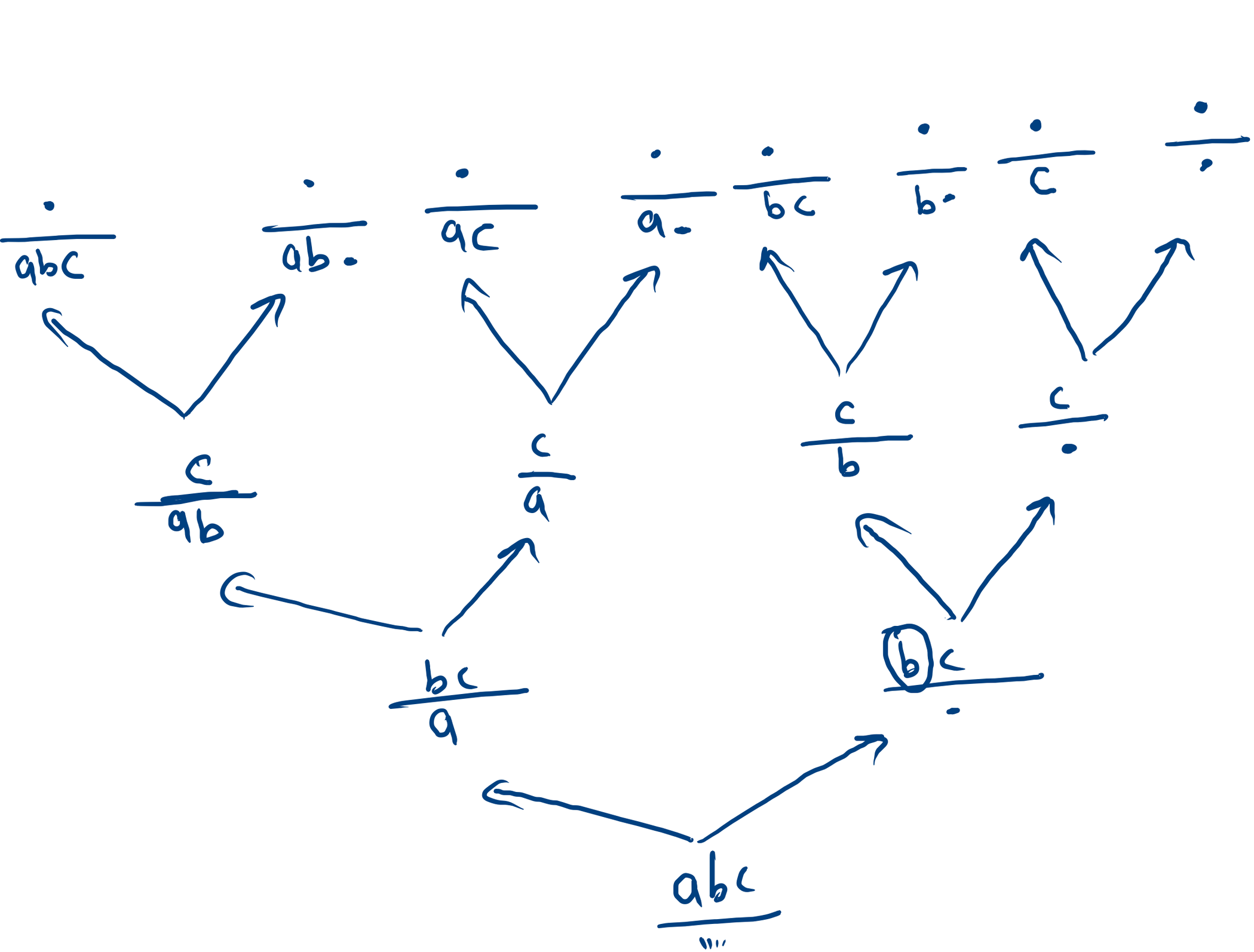
yvTA
yvT
yvA
yv
yTA
yT
yA

yvTA

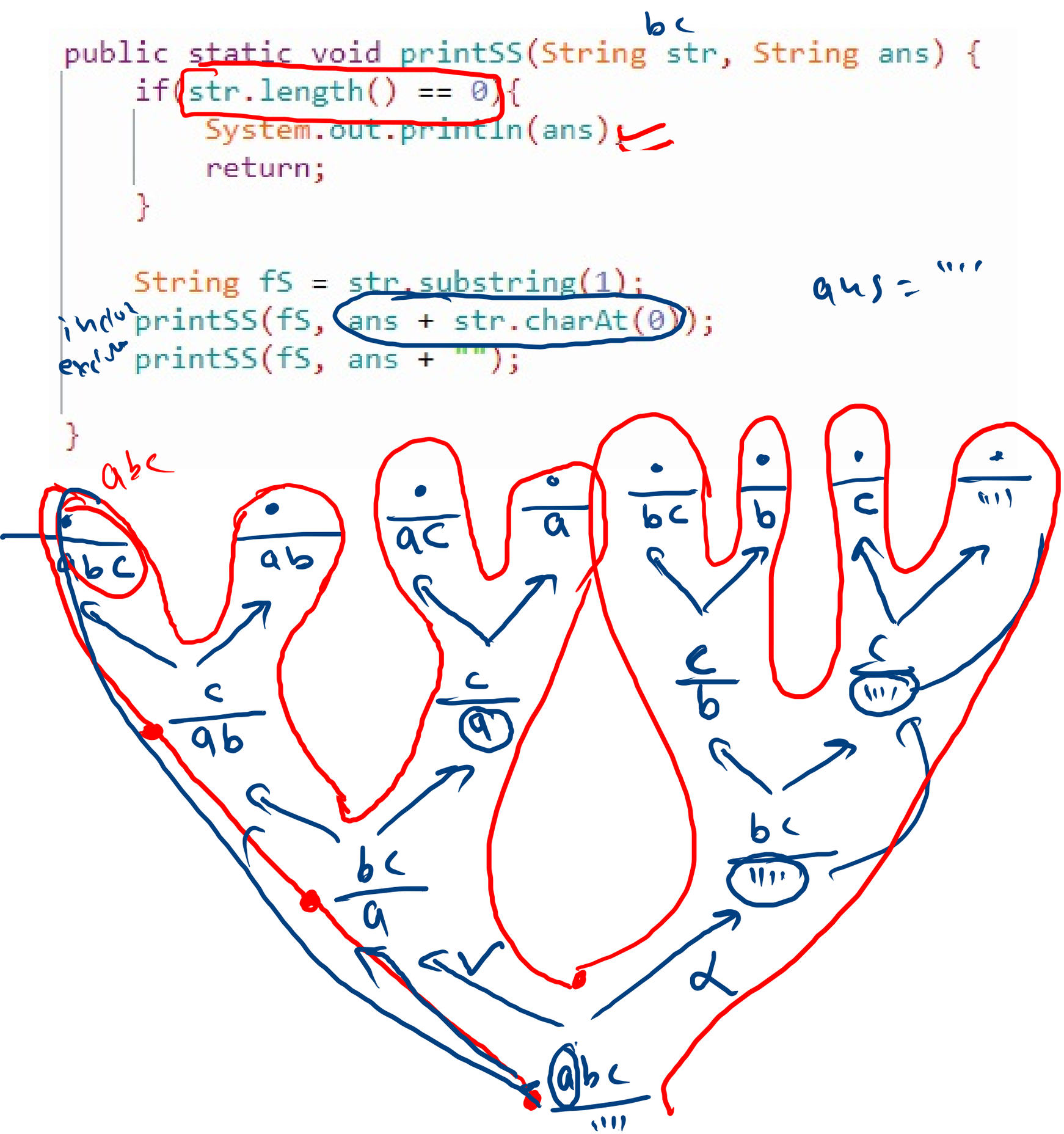
Sample C

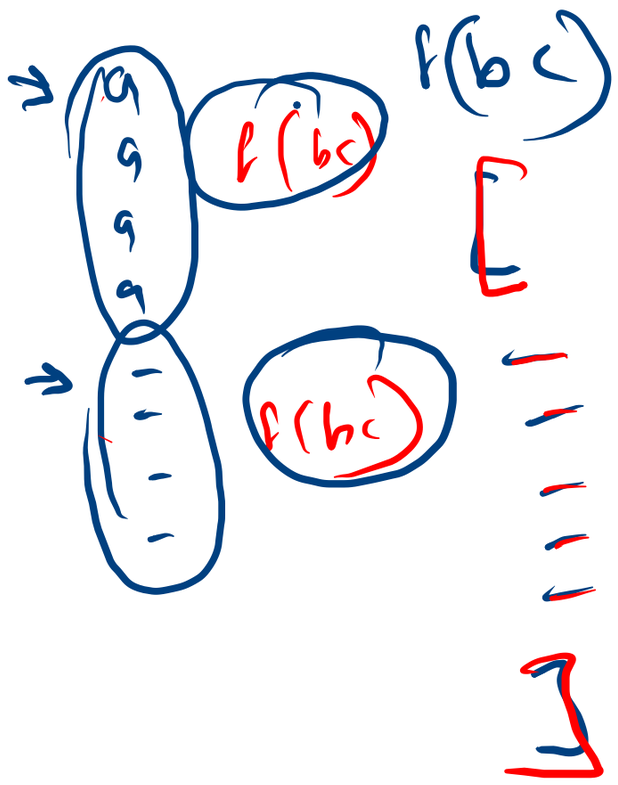
yvTA
yvT
yvA
yv
yTA
yT
yA


str → abc
abc
ab
ac
a
bc
b
c
fail bc
a fail
fail
Arraylist


abc
bc
a
ab
ac
a
bc
b
c
b
c
c
b
c
c
abc

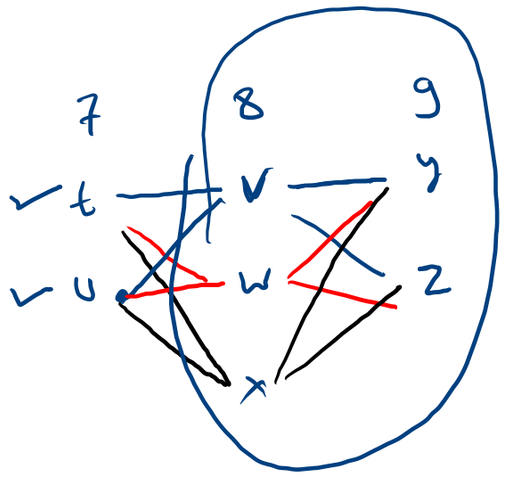
```java
public static void printSS(String str, String ans) {
    if(str.length() == 0){
        System.out.println(ans);
        return;
    }

    String fS = str.substring(1);
    printSS(fS, ans + str.charAt(0));
    printSS(fS, ans + "");

}
```



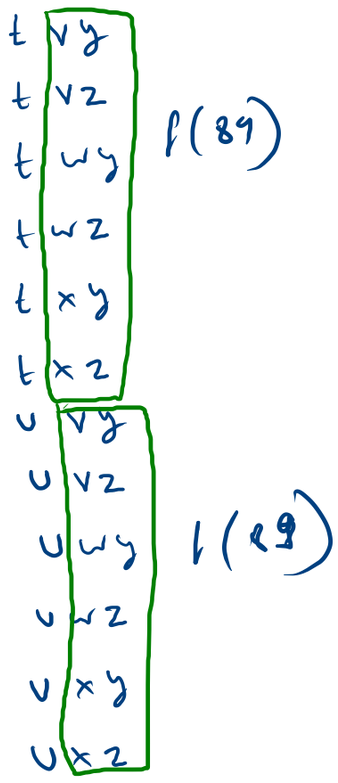


0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

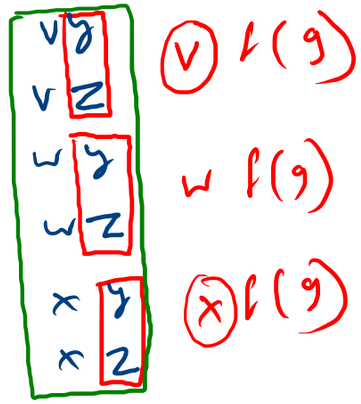
7 8 9



7 8 9



f(8 9)



8 -> v w x

9 -> y z

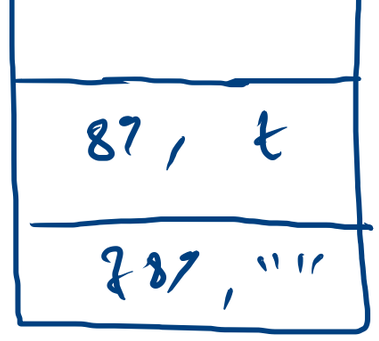
f(9)

y
z

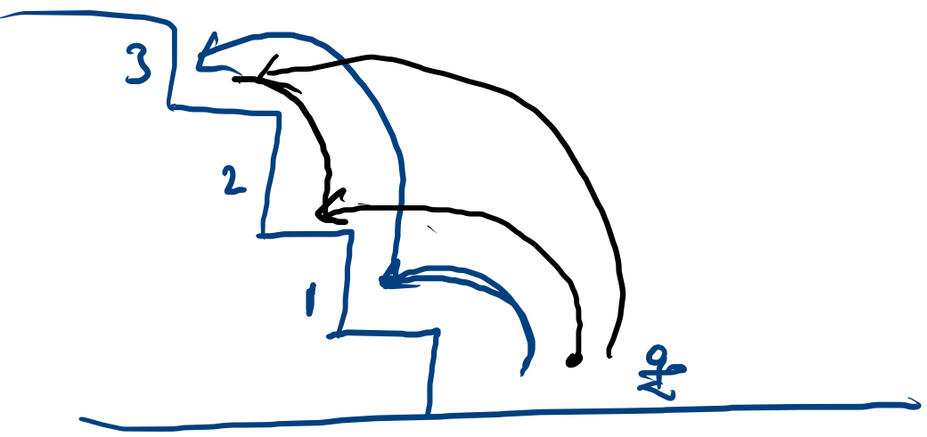
higher Level

```java
public static void printKPC(String str, String asf) {
    if(str.length() == 0){
        System.out.println(asf);
        return;
    }

    int index = str.charAt(0) - '0';
    String code = codes[index];

    String fS = str.substring(1);
    for(int i=0;i<code.length();i++){
        char cod = code.charAt(i);
        printKPC(fS, asf+cod);
    }
}
```

0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz

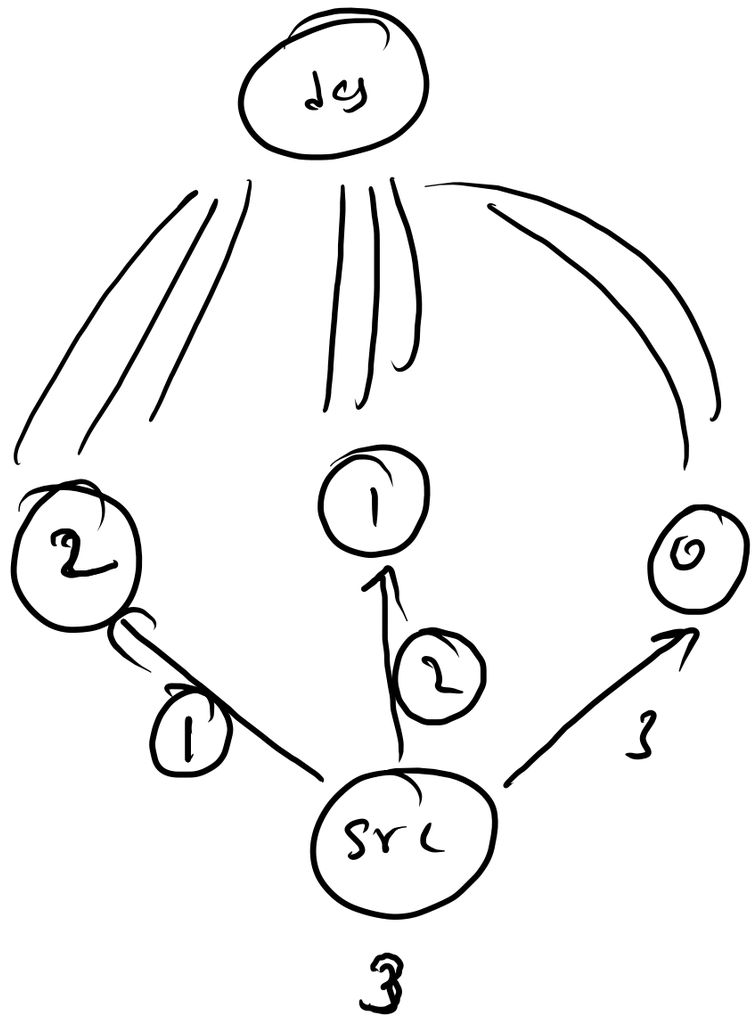


n=3



jump [1, 2, 3]

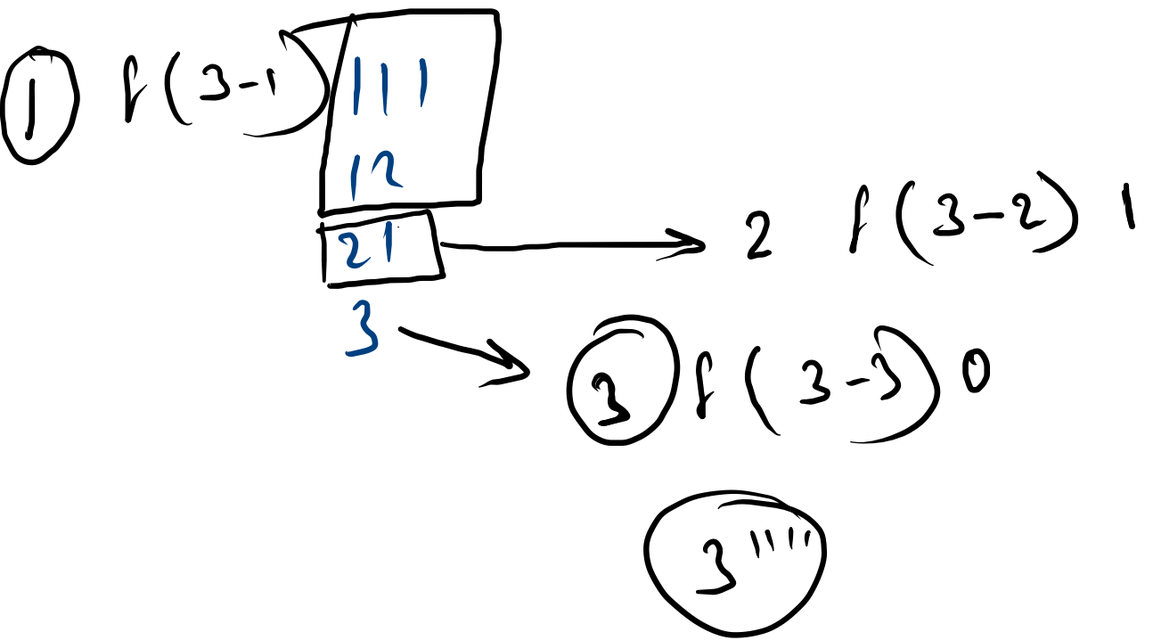
1	1
1	2
2	1
3	



ss Vd

klc code loop code

n = 3



2 1 0

'''' '''

11 1

2

21

10pts

20pts

30pts

3-1

2 → 3-2 → 1

3 → 3-1 = 0

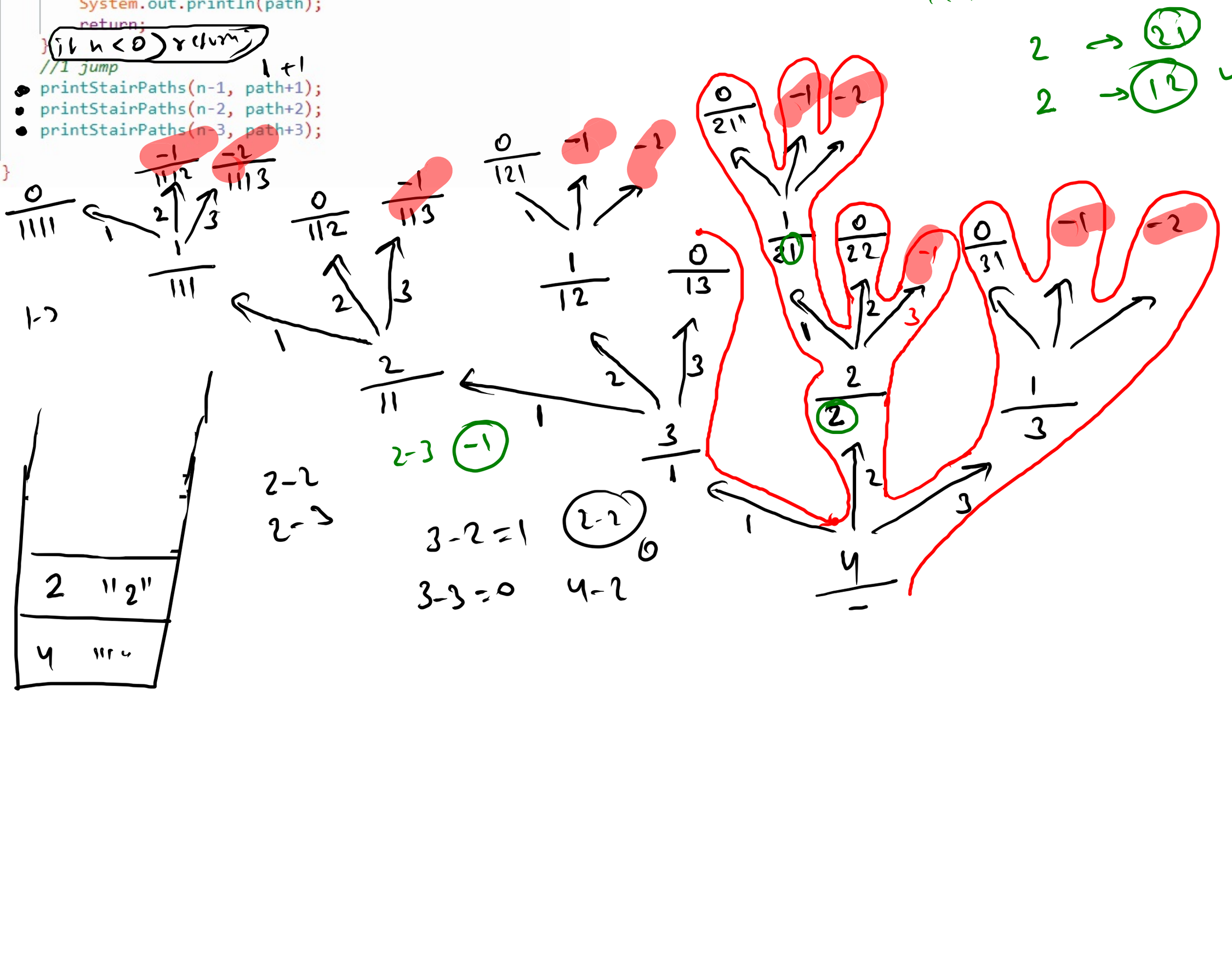
```java
public static void printStairPaths(int n, String path) {
    if(n==0){
        System.out.println(path);
        return;
    }
    //1 jump
    printStairPaths(n-1, path+1);
    printStairPaths(n-2, path+2);
    printStairPaths(n-3, path+3);

}
```

if n<0 return

1+1

ans+1
1+ans

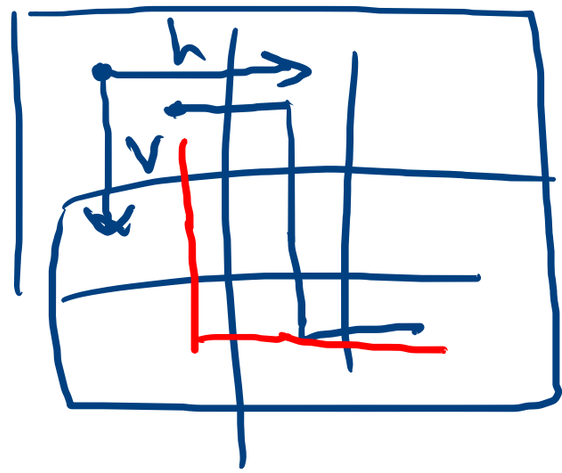
2 → 21
2 → 12 ✓



h = 3

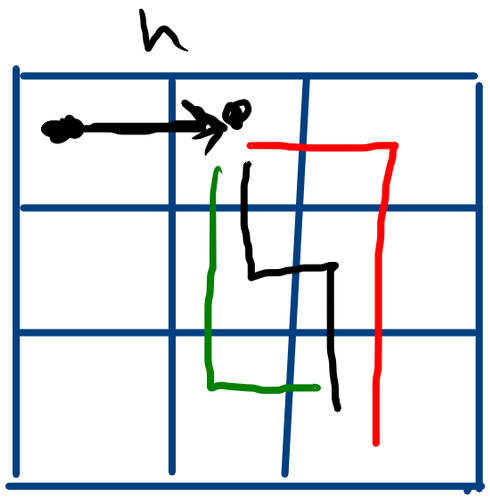
m = 3

SS



hvvh

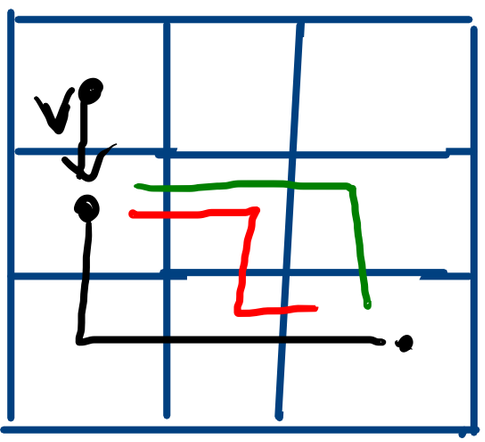
vvhh



$f(r,c) \rightarrow$ h $f(r, c+1)$

h
h
h

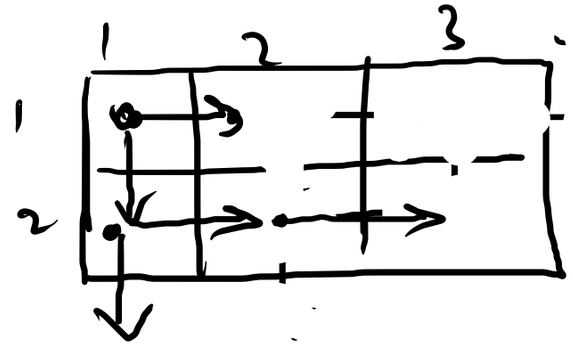
hvv
v hv
vvh



$f(r,c) \rightarrow$ v $f(r+1, c)$ /v

```java
public static void printMazePaths(int sr, int sc
    if(sr==dr && sc == dc){
        System.out.println(psf);
    }

    // horizontal
    if(sc+1<=dc)
    printMazePaths(sr, sc+1, dr, dc, psf+"h");

    if(sr+1 <= dr)
    printMazePaths(sr+1, sc, dr, dc, psf+"v");

}
```

q
b



h = 2

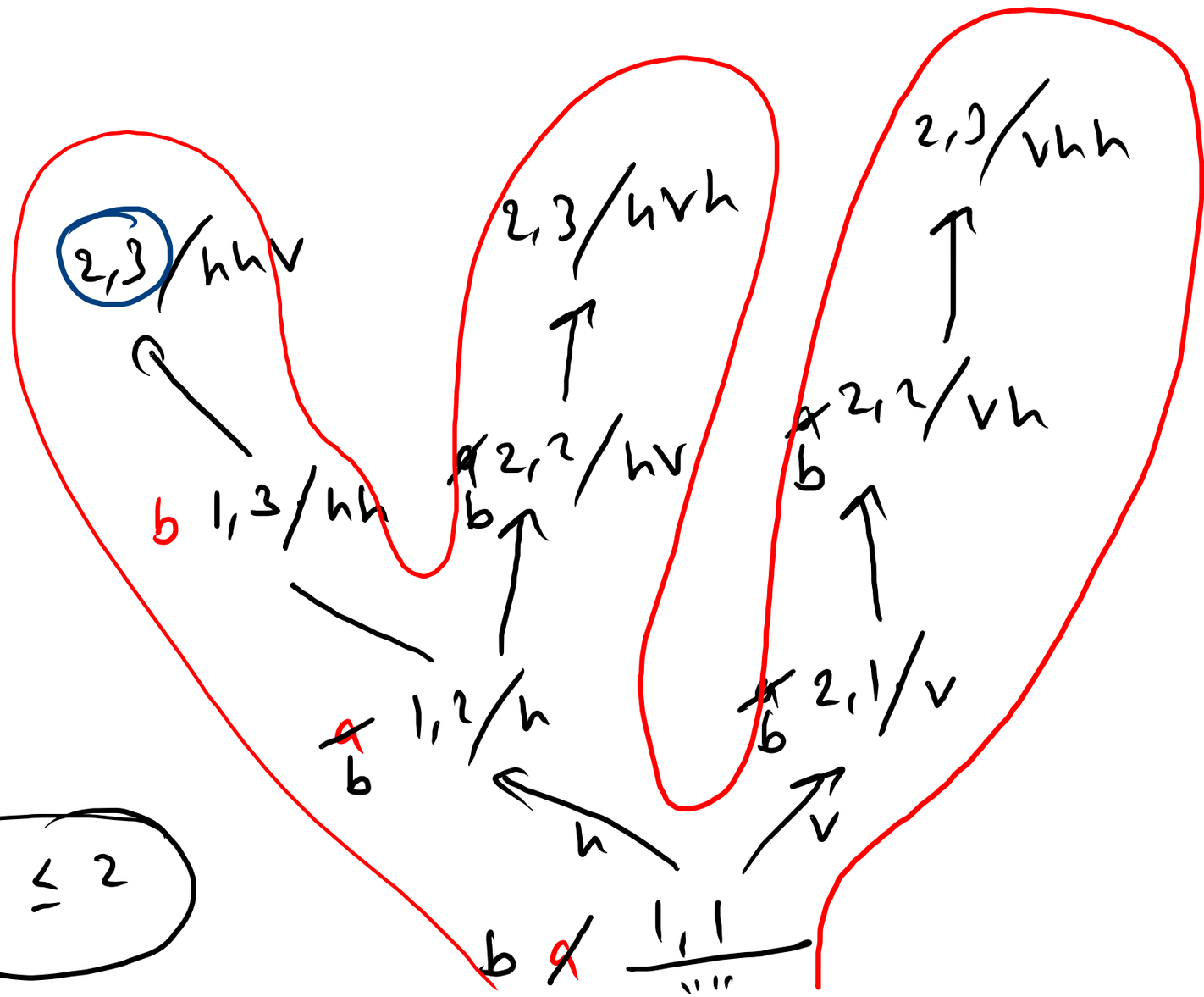
m = 3

3+1 ≤ 3

2+1 ≤ 2

hhv
hvh
vhh

2+1 ≤ 2



Str → a b c
0 1 2

Permutation

[a b c
  a c b

b [b a c
  b c a

[c a b
  c b a

a f(bc) < bc / cb

b f(ac) < ac / ca

c f(ab) < ab / ba

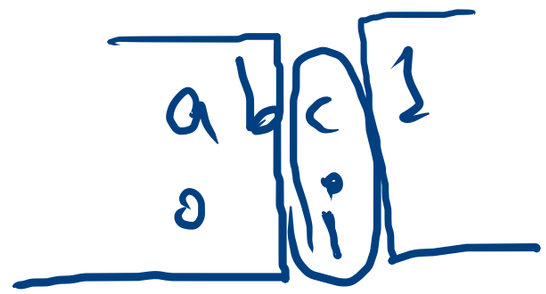
Sh → abcd
0123

[a f(bcd)

[b f(acd)

[c f(abd)

[d f(abc)



abd

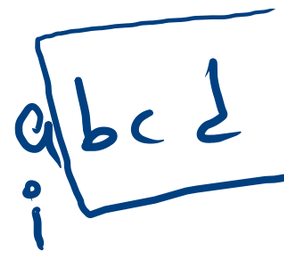
ss(0,i) + ss(i+1

ab

2m

d

1m



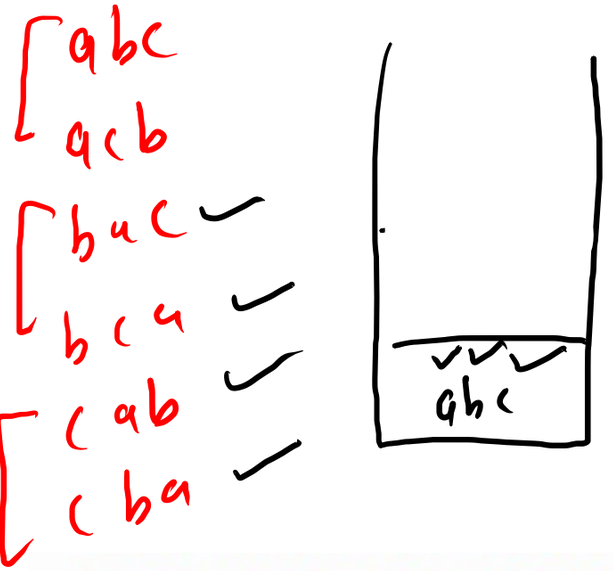
→ bcd

ss(0,i) + ss(i+1)

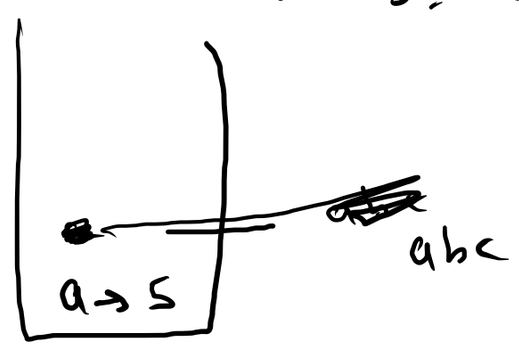
1111

bcd

```java
public static void printPermutations(String str, String asf) {
    if(str.length() == 0){
        System.out.println(asf);
    }

    for(int i=0;i<str.length(); i++){
        char ch = str.charAt(i);
        String fS = str.substring(0, i) + str.substring(i+1);
        printPermutations(fS, asf+ch);
    }
}
```




```java
public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);
    printPermutations(scn.nextLine(), "");
}
```



ch

a → 1

b → 2

c → 3

.

.

z → 26

z → 26

'a' - 'a' +1

97 - 97 = 0 1

'b' - 'a' +1

98 - 97 = 1 = 2

int val = ch - 'a' + 1

sh → 1 2

ans → a b