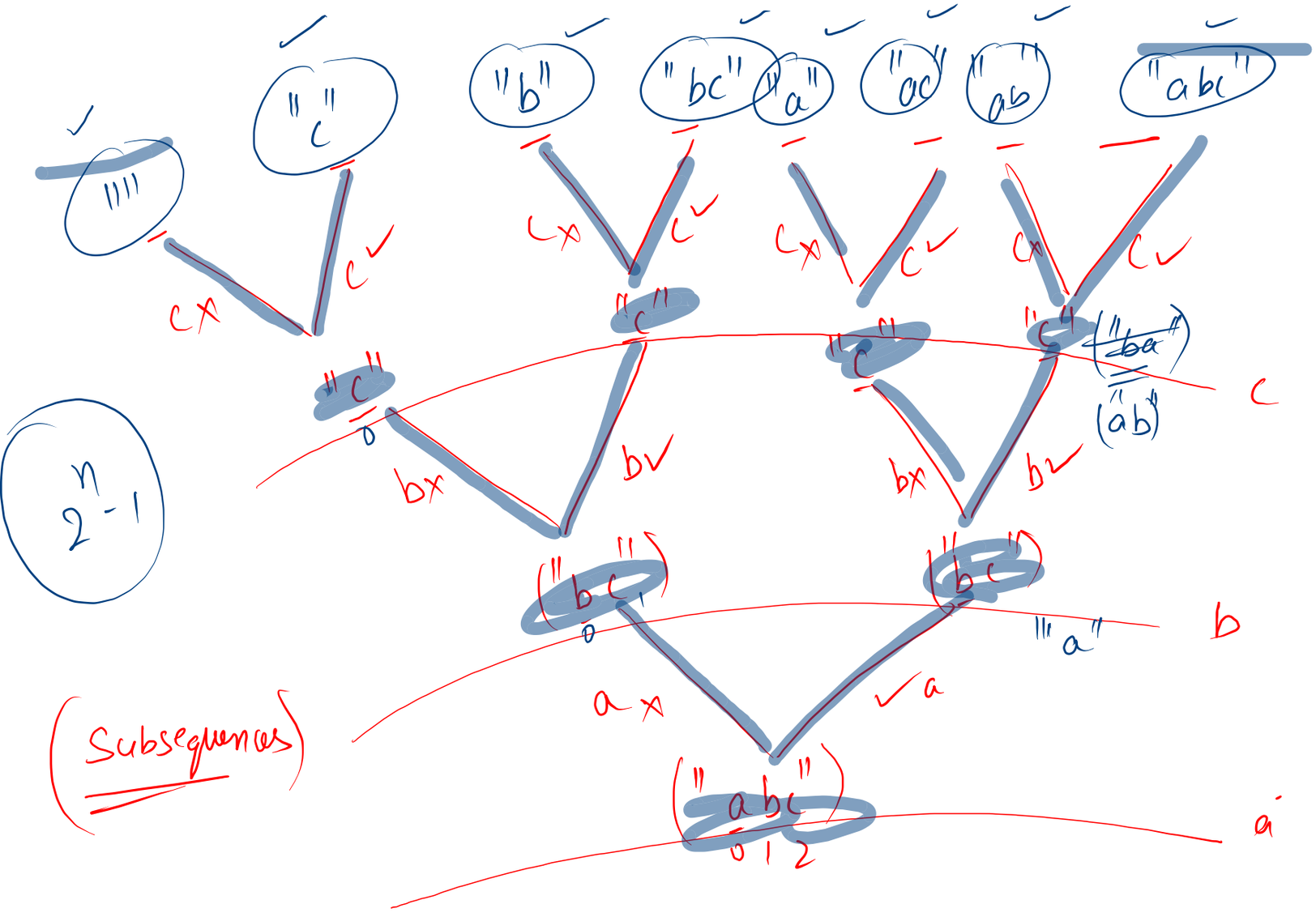
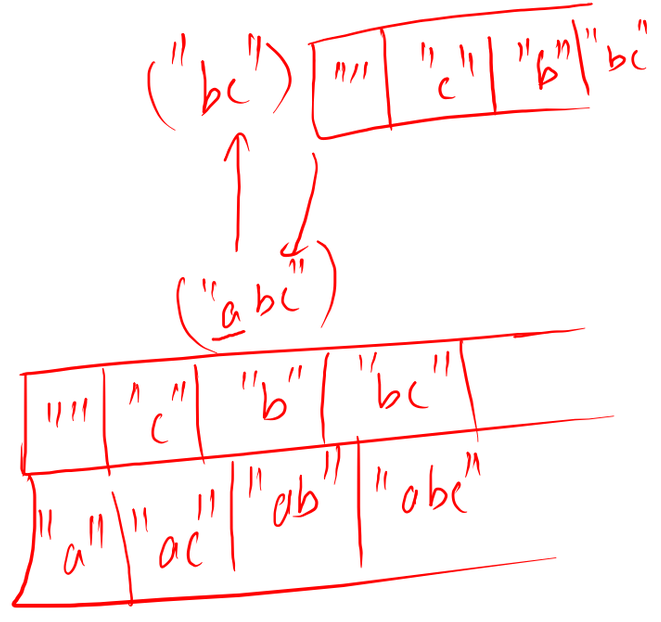
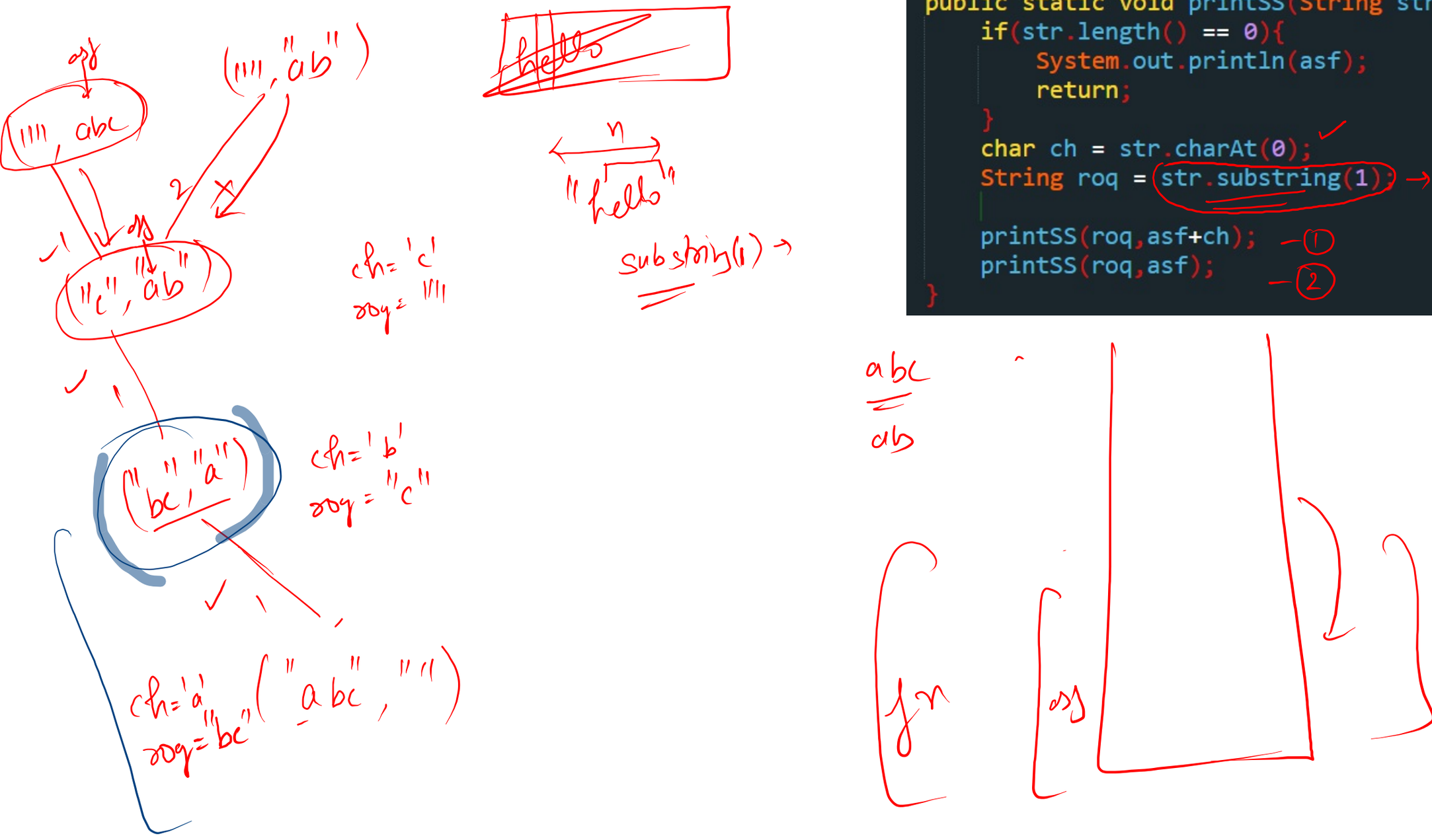
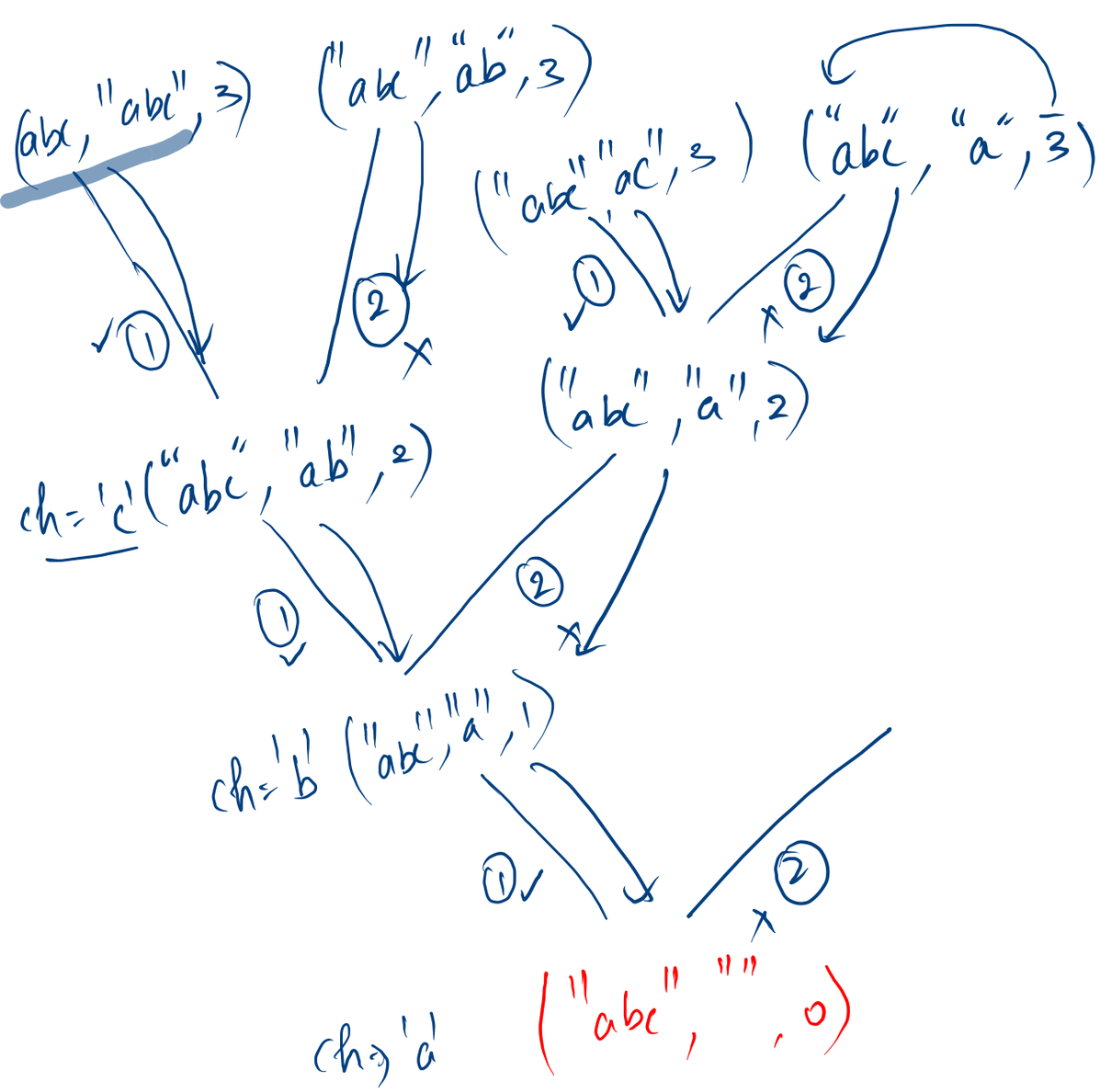



```java
public static void printSS(String str, String asf) {
    if(str.length() == 0){
        System.out.println(asf);
        return;
    }
    char ch = str.charAt(0);
    String roq = str.substring(1);

    printSS(roq,asf+ch);
    printSS(roq,asf);
}
```



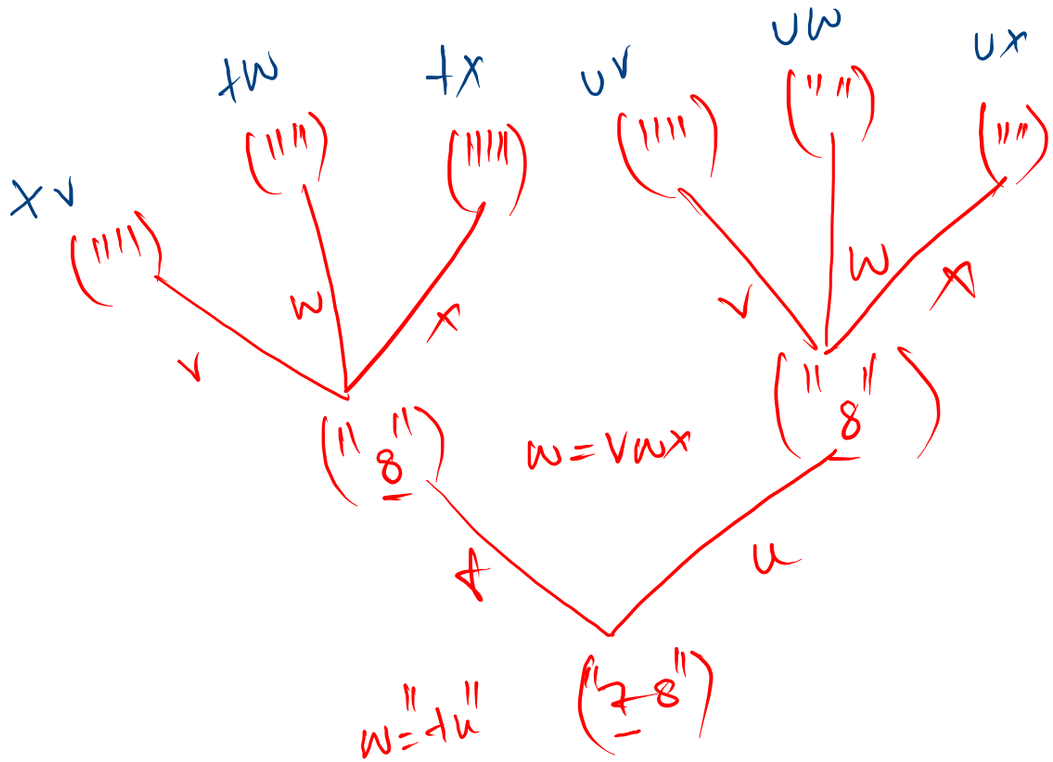


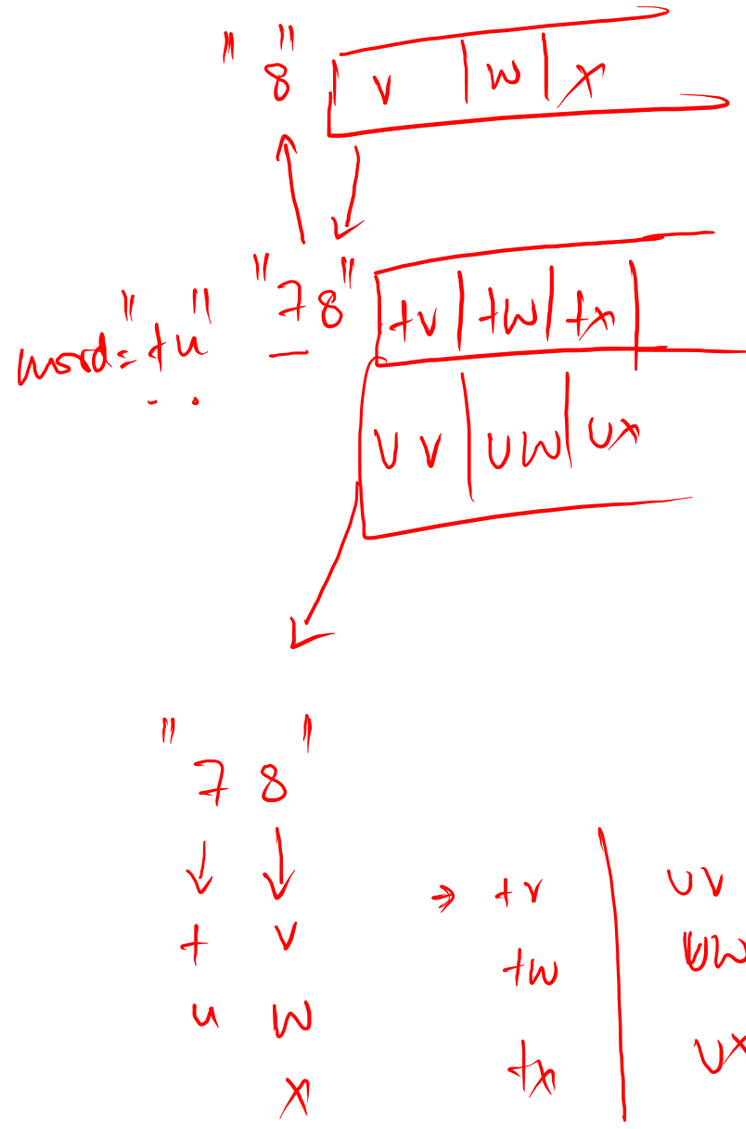

```java
public static void printSS(String str, String asf) {
    if(str.length() == 0){
        System.out.println(asf);
        return;
    }
    char ch = str.charAt(0);
    String roq = str.substring(1);

    printSS(roq,asf+ch);
    printSS(roq,asf);
}
```




```java
public static void printSS(String str, String asf,int idx) {
    if(idx == str.length()){
        System.out.println(asf);
        return;
    }
    char ch = str.charAt(idx);

    printSS(str,asf+ch,idx+1); - 1
    printSS(str,asf,idx+1);    - 2
}
```

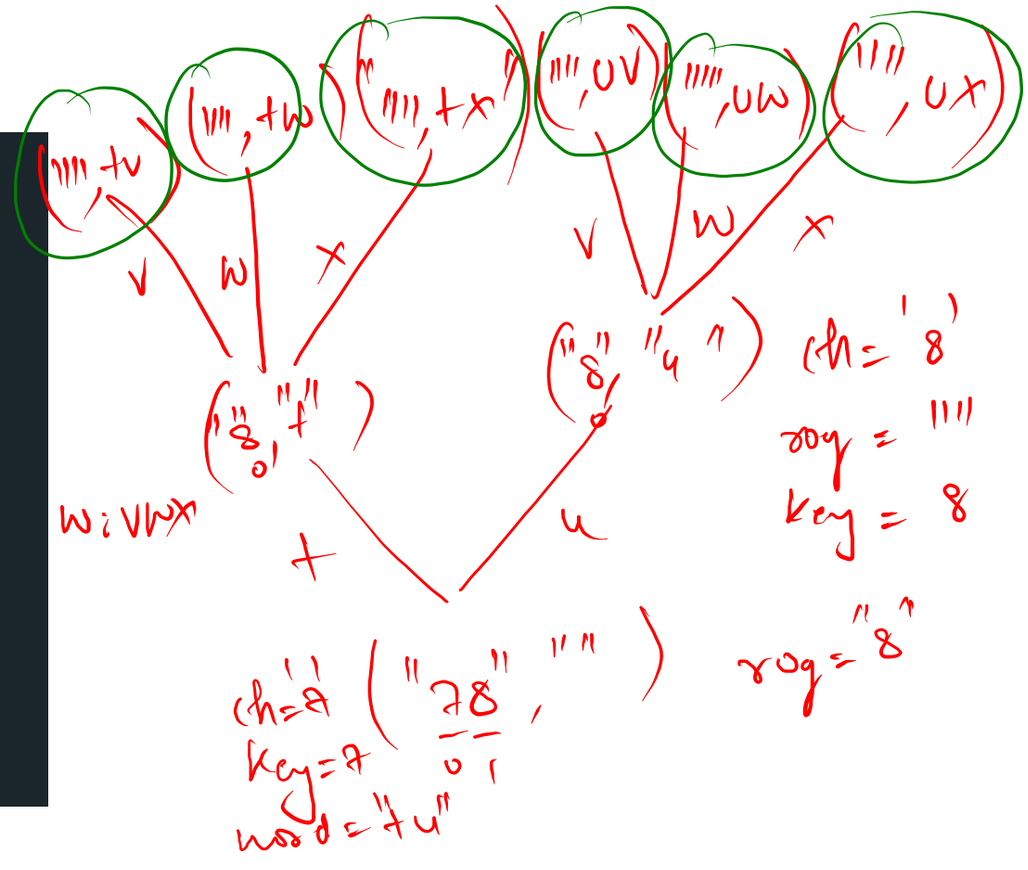
0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> tu
8 -> vwx
9 -> yz



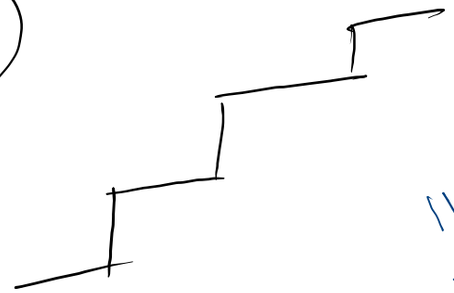


0 1 2 3 4 5 6 7 8 9

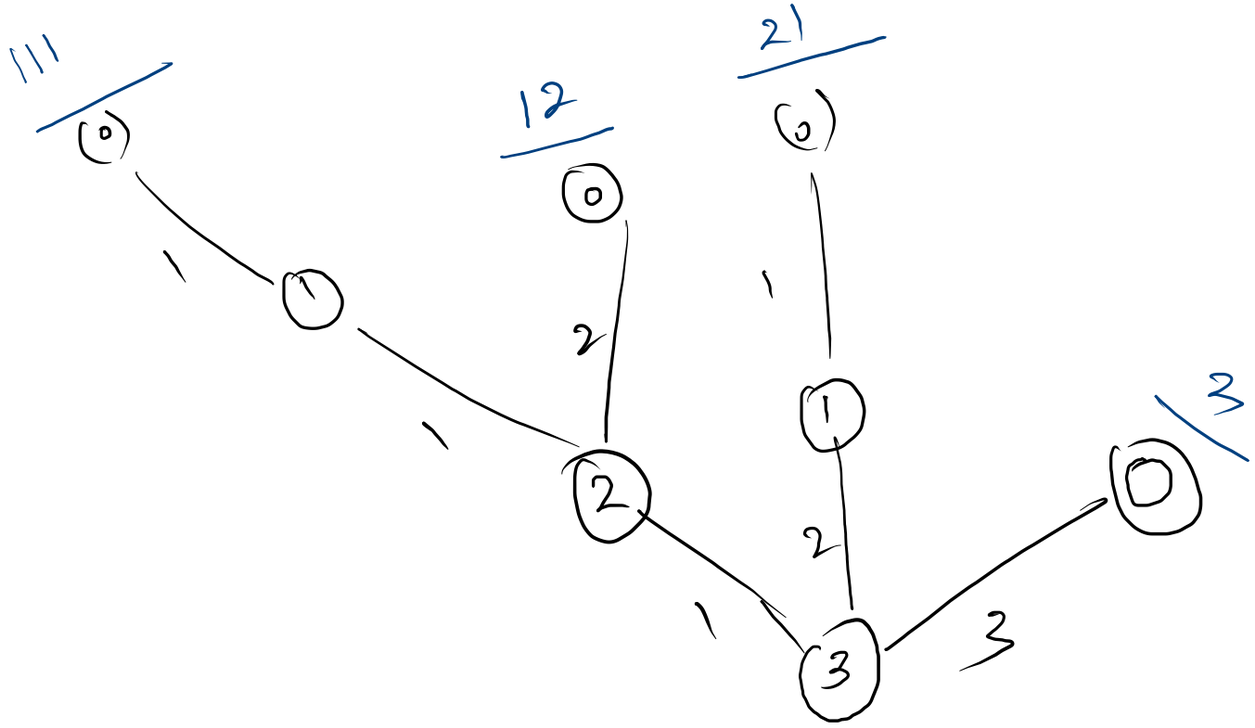
```java
static String keypad[] = {".;","abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};

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

    char ch = str.charAt(0);
    String roq = str.substring(1);

    int key = Integer.parseInt(ch+"");
    String word = keypad[key];

    for(int i = 0 ; i < word.length() ; i++){
        printKPC(roq,asf+word.charAt(i));
    }
}
```



n = 3



1 len, 2 len, 3 len,

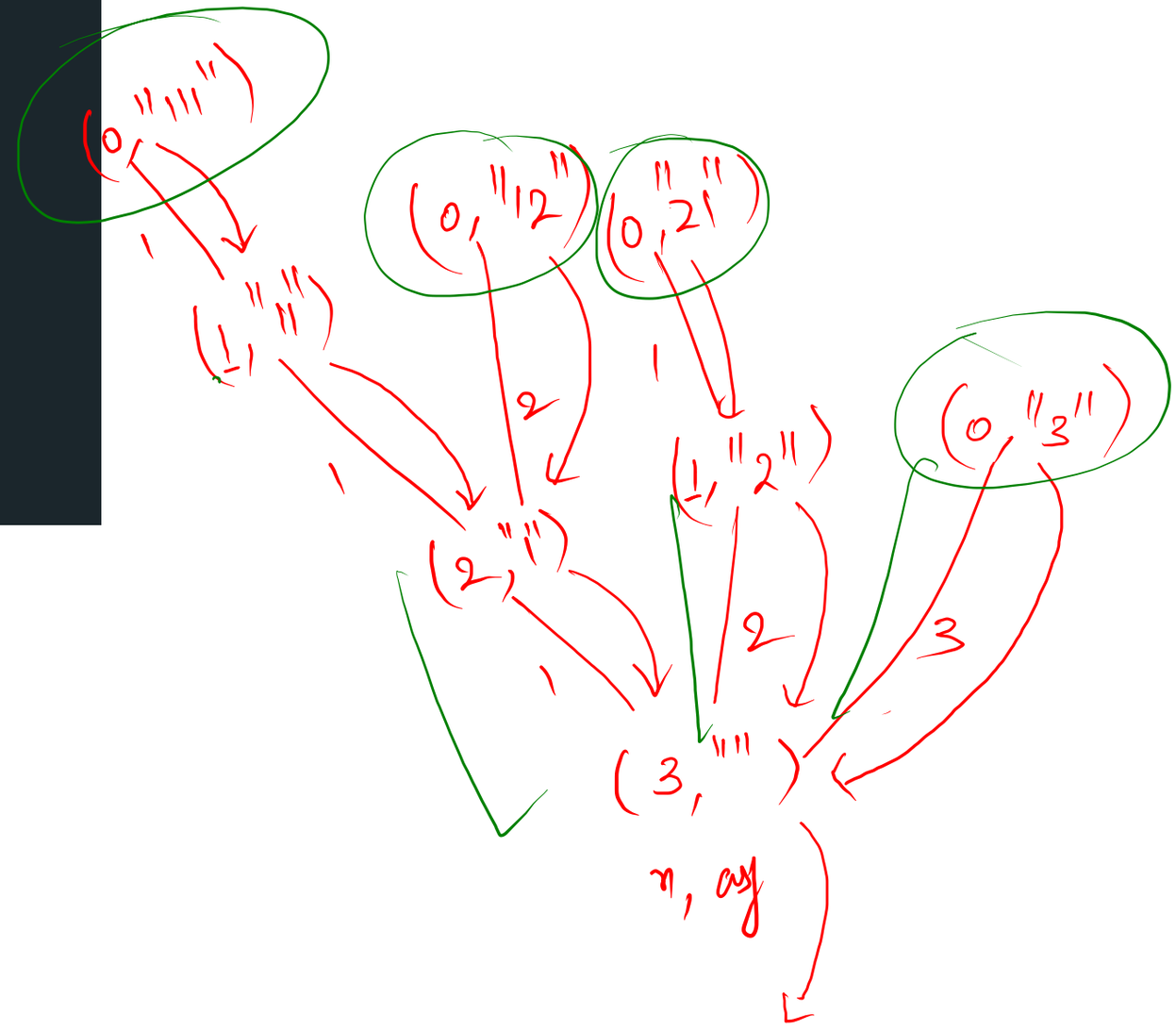


```java
public static void printStairPaths(int n, String psf) {
    if(n == 0){
        System.out.println(psf);
        return;
    }
    if(n-1 >= 0){
        printStairsPaths(n-1,psf+"1");
    }
    if(n-2 >= 0){
        printStairsPaths(n-2,psf+"2");
    }
    if(n-3 >= 0){
        printStairsPaths(n-3,psf+"3");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

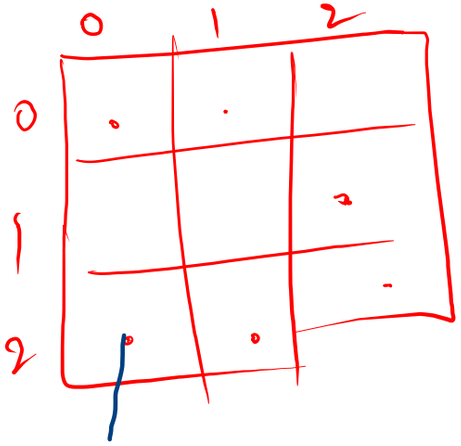
→ Working

→ Proactive

→ Team



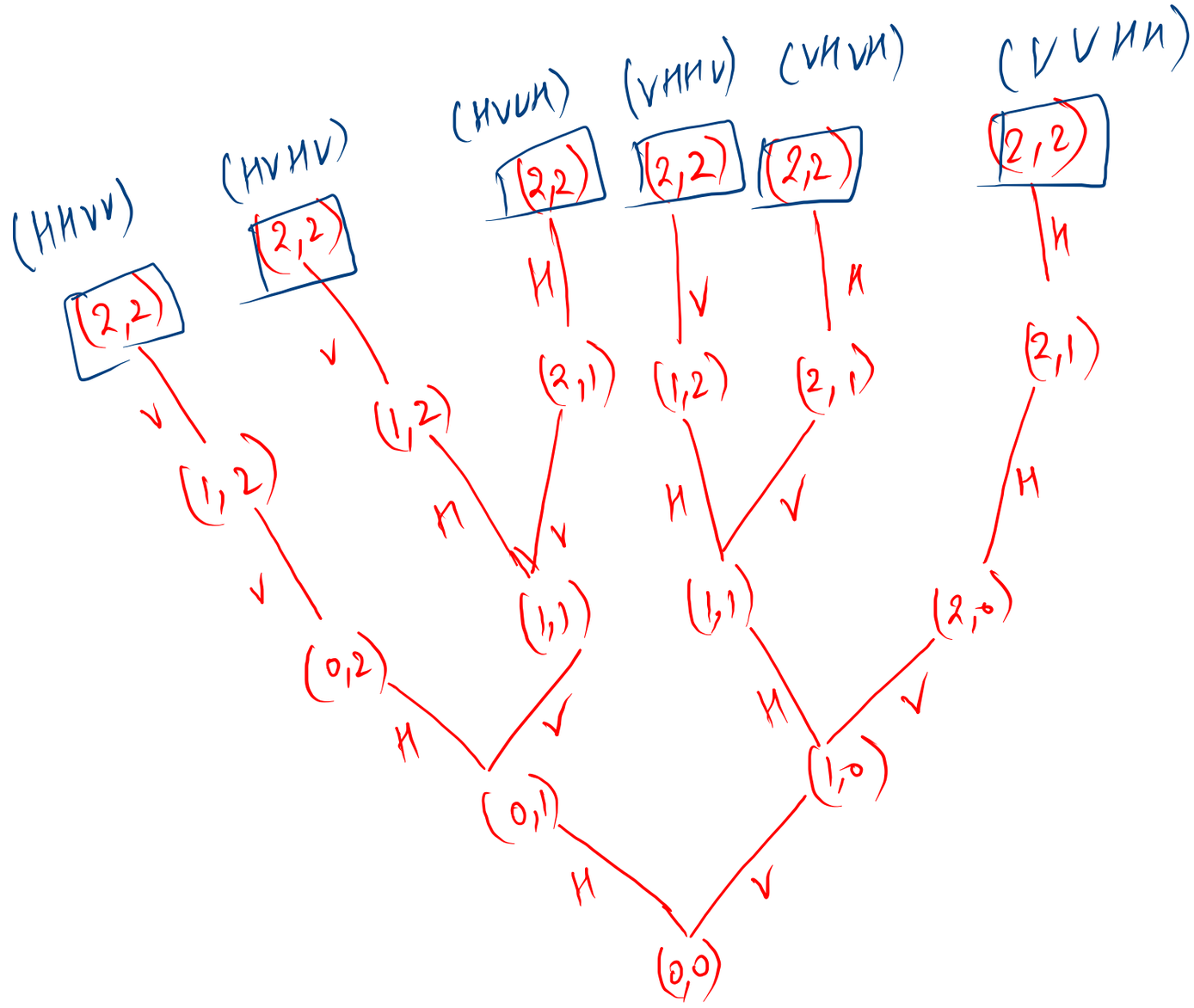
111
12
21
3

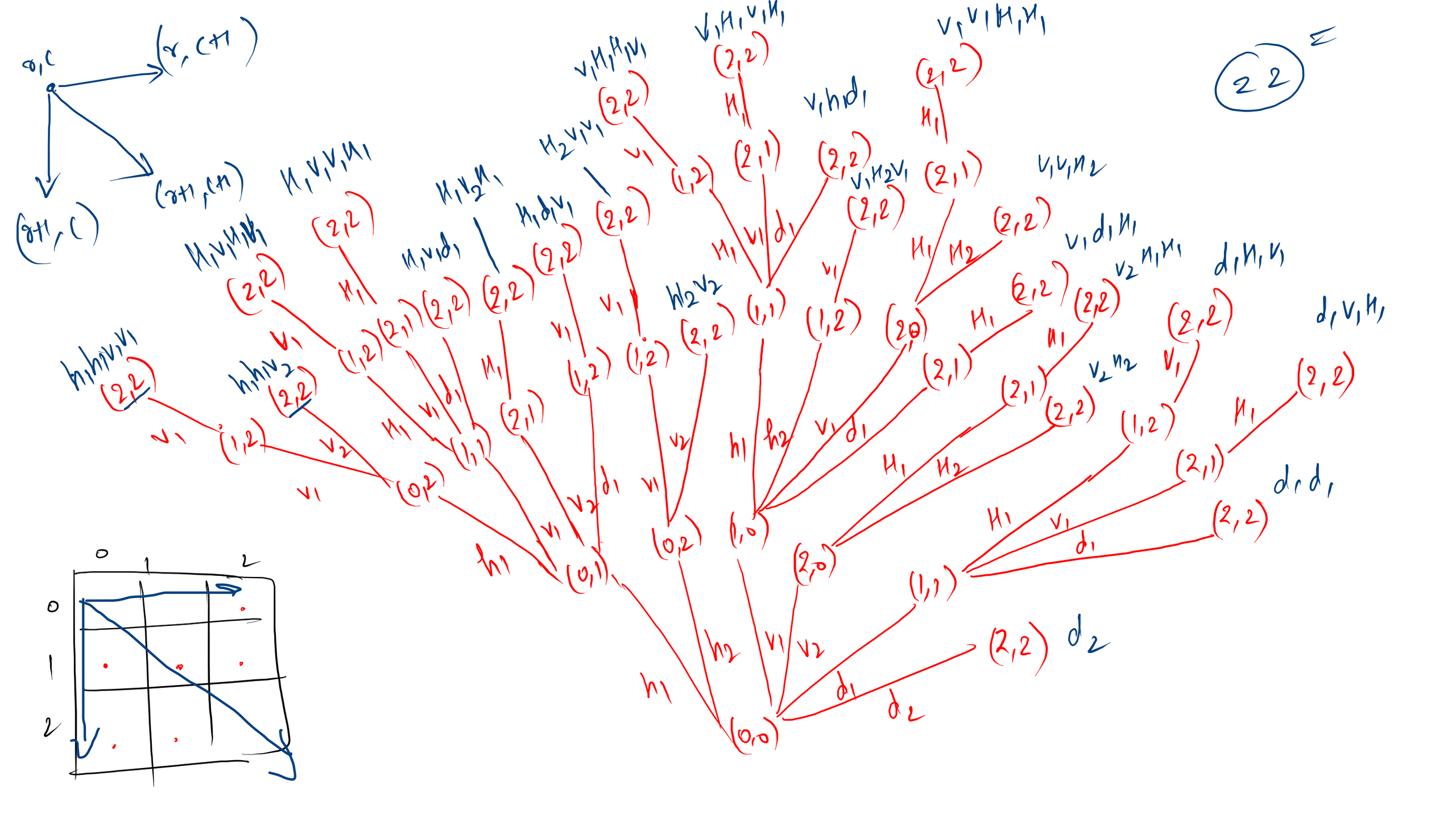


$(r, c) \xrightarrow{H} (r, c+1)$

$\downarrow V$

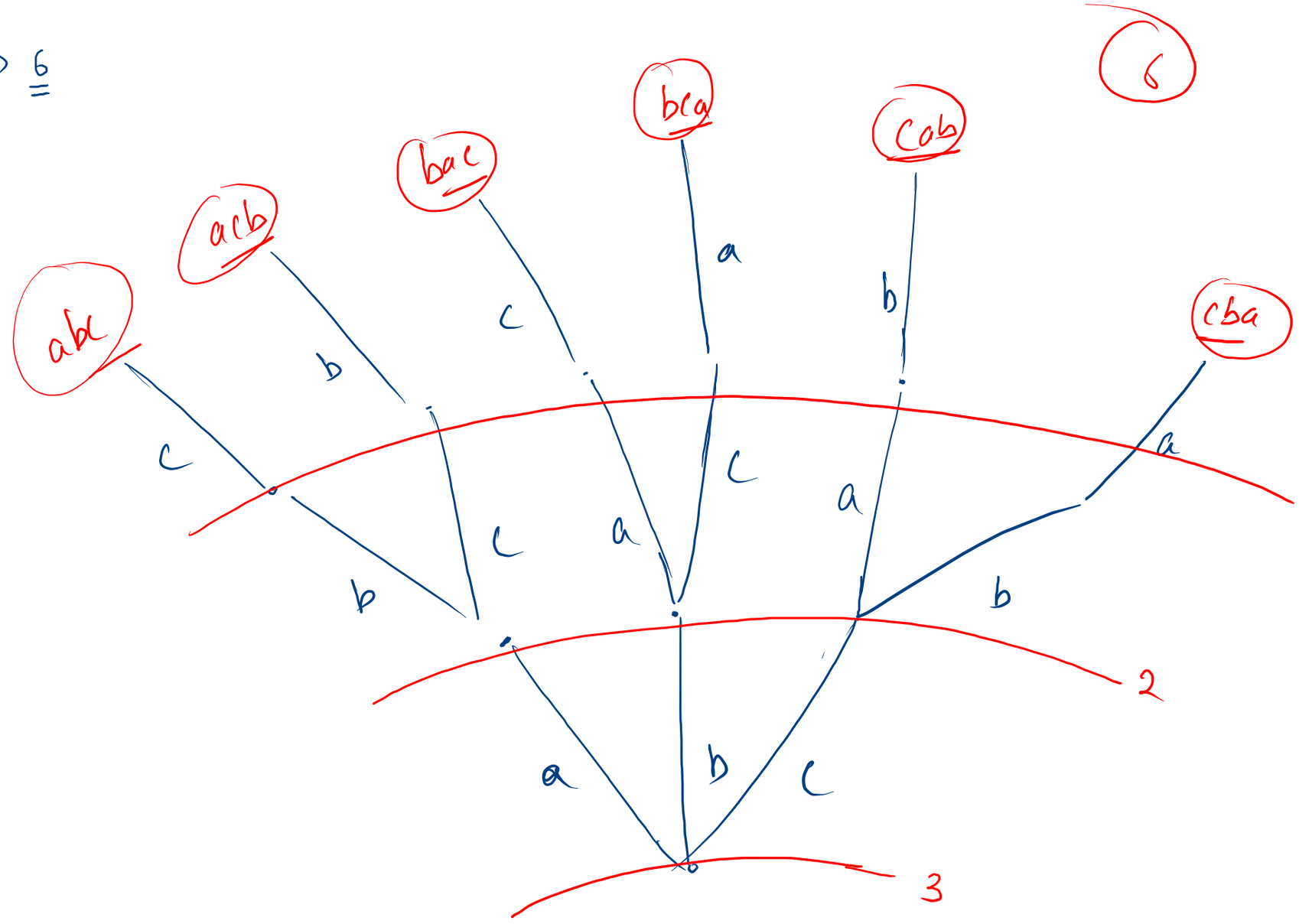
$(r+1, c)$

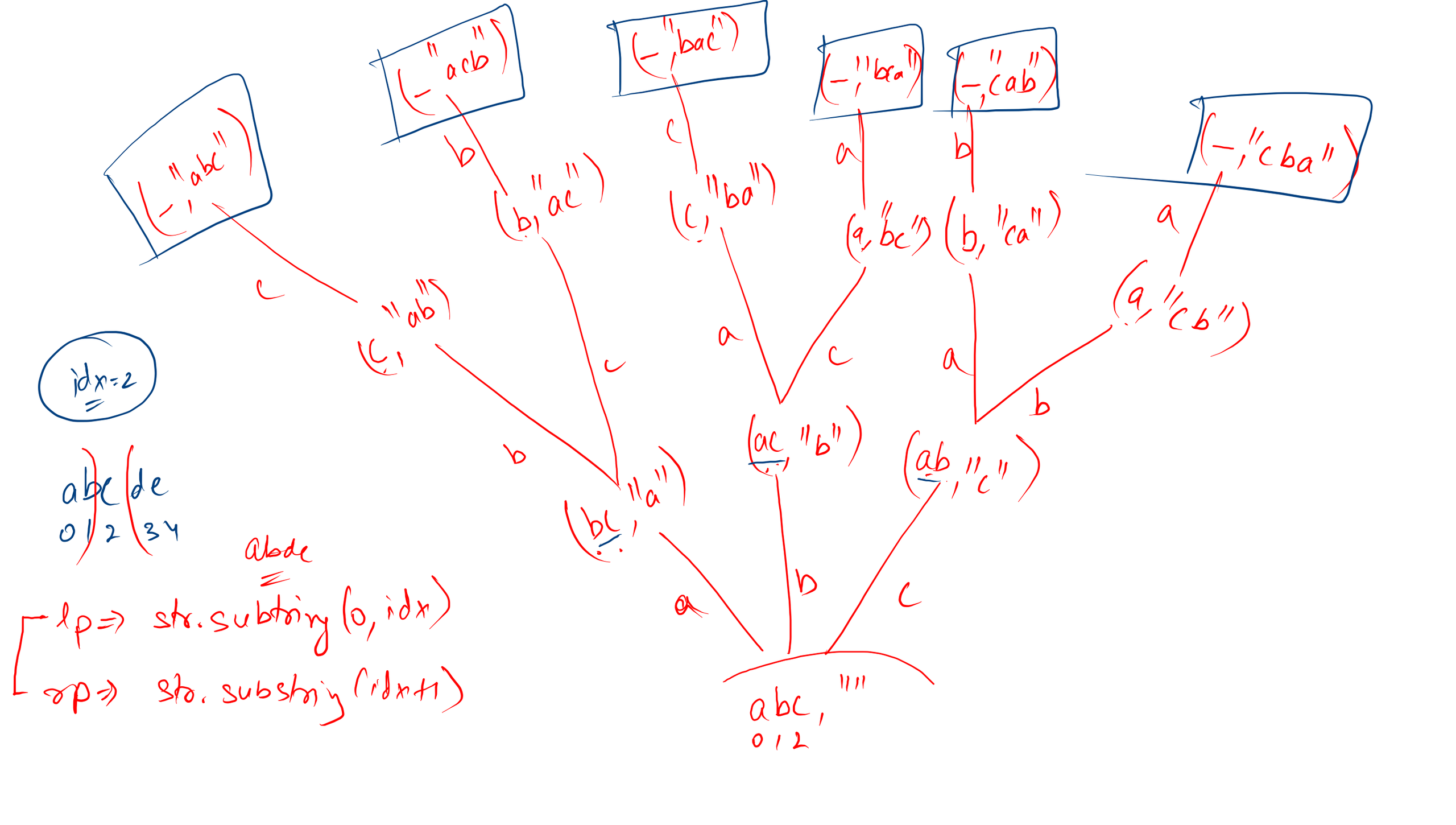


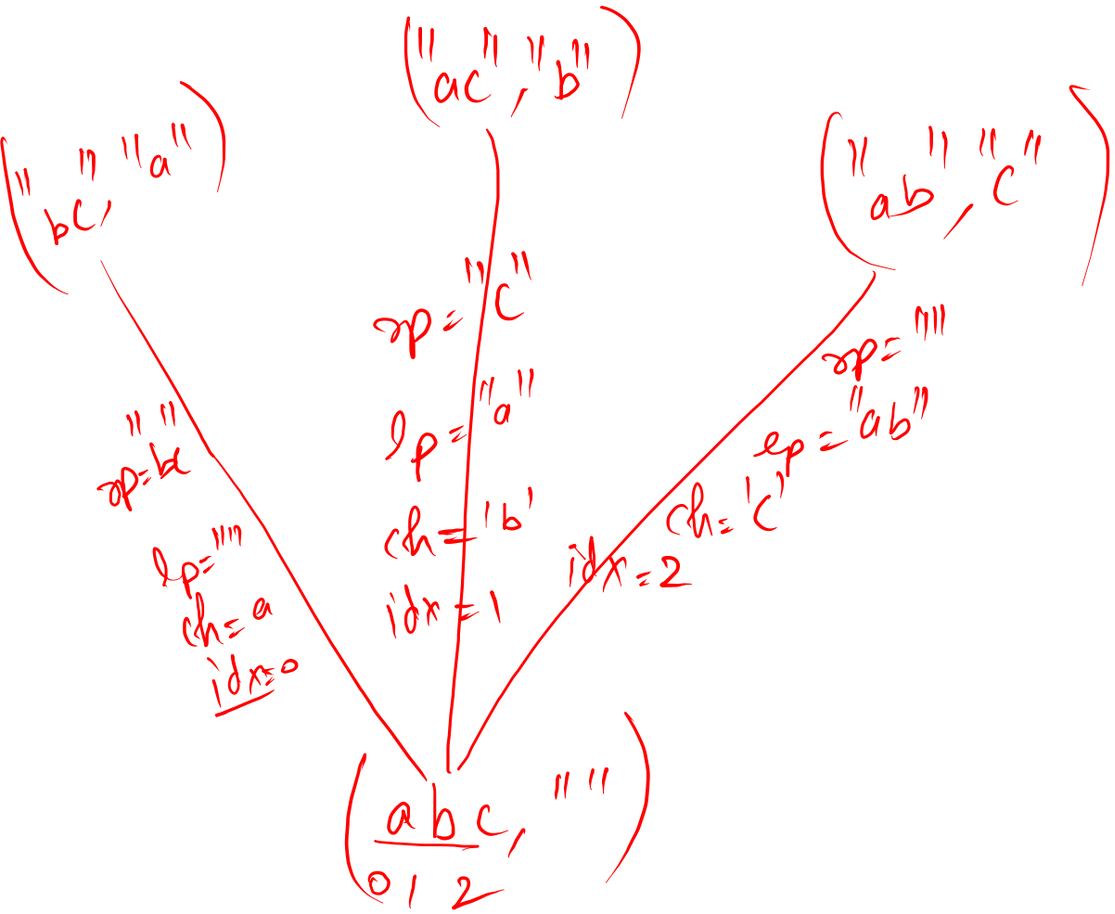

(r, c+1)
(r+1, c+1)
(r+1, c)
22

$[abc] \rightarrow 3\ 2\ 1 \Rightarrow 6$

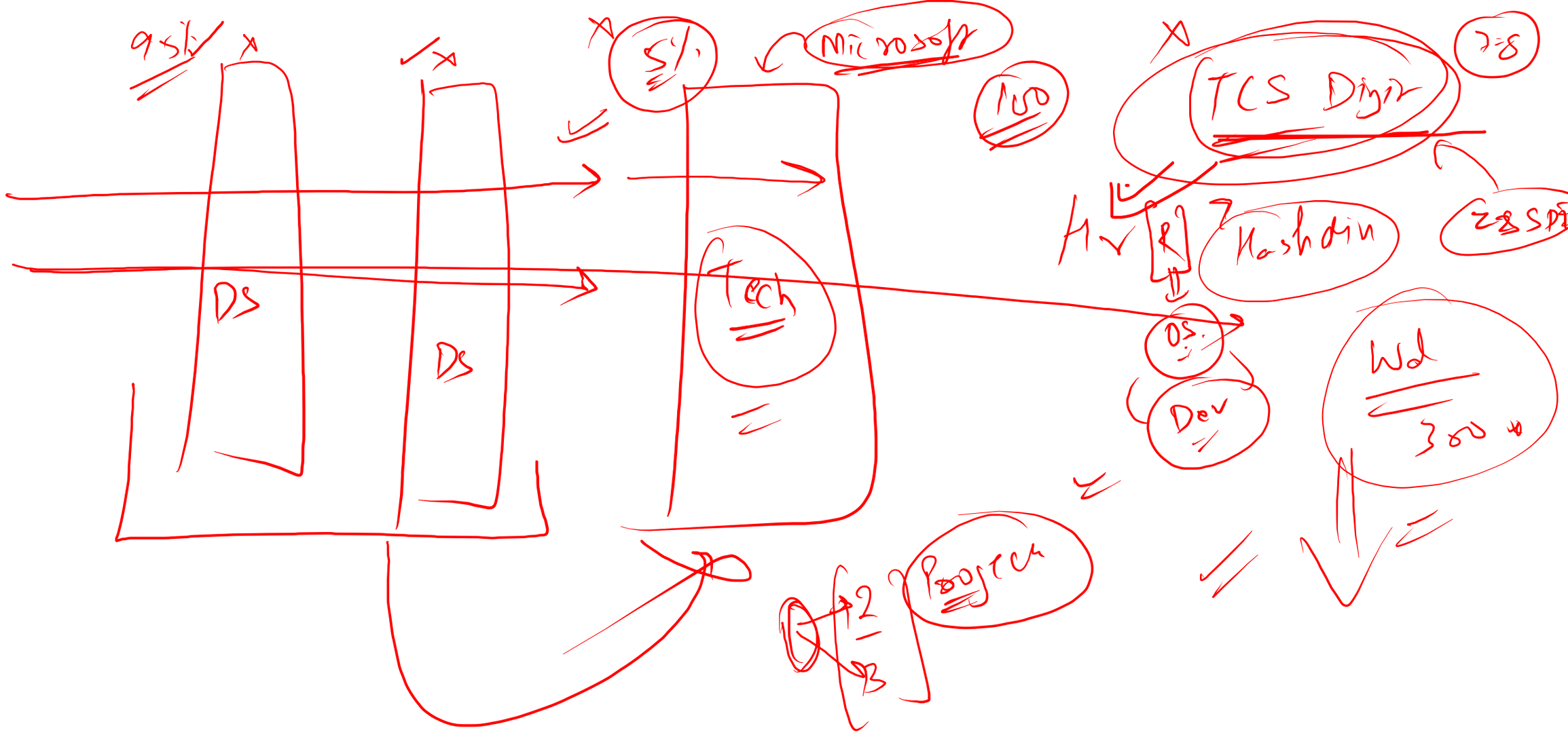
$$\begin{bmatrix} abc \\ acb \\ bac \\ bca \\ cab \\ cba \end{bmatrix}$$




(-, "abc")
(-, "acb")
(-, "bac")
(-, "bca")
(-, "cab")
(-, "cba")
c
b
c
a
b
a
("ab", c)
(b, "ac")
(c, "ba")
(a, "bc")
(b, "ca")
(a, "cb")
a
c
c
a
b
a
b
("a", bc)
(ac, "b")
(ab, "c")
a
b
c
("", abc)
idx=2
abcde
0 1 2 3 4
abcde
lp => str.substring(0, idx)
rp => str.substring(idx+1)




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

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


DS
DS
Tech
ISO
TCS Digit
Hashdin
DS
Dev
Wd
300
Project