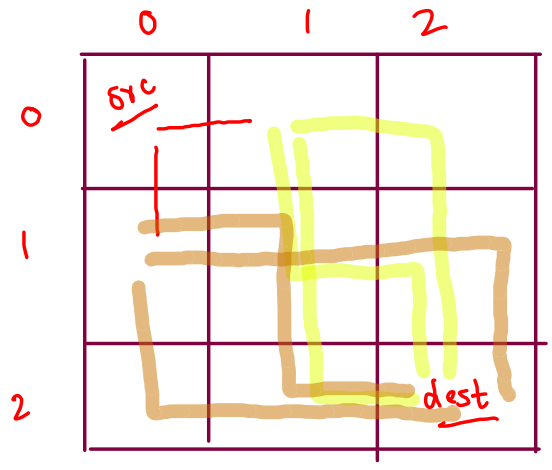
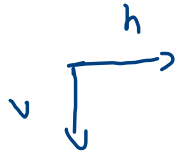
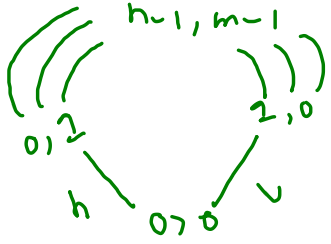




0,1 to dest → hvv, vvh, vhv

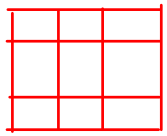
0,0 to dest → hhvv, hvvh, hvhv



1,0 to dest → hhv, hvh, vhh

0,0 to dest → vhhv, vhvh, vvhh

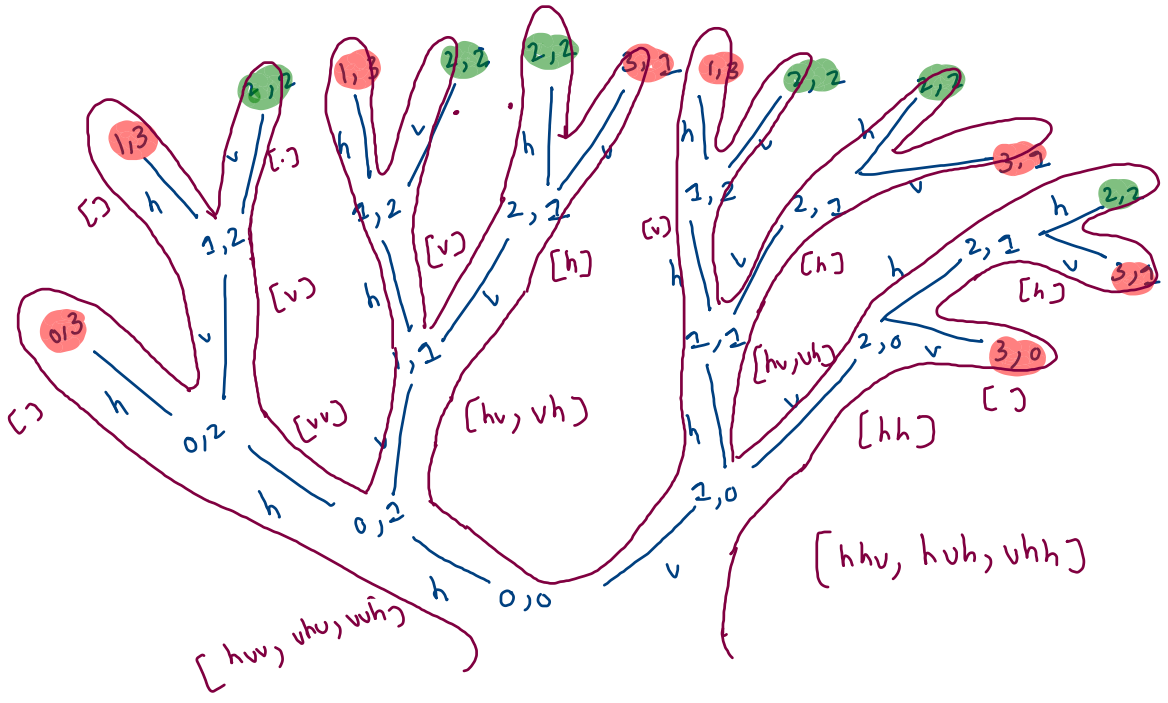
0,0 to dest → hhvv, hvvh, hvhv, vhhv, vhvh, vvhh.

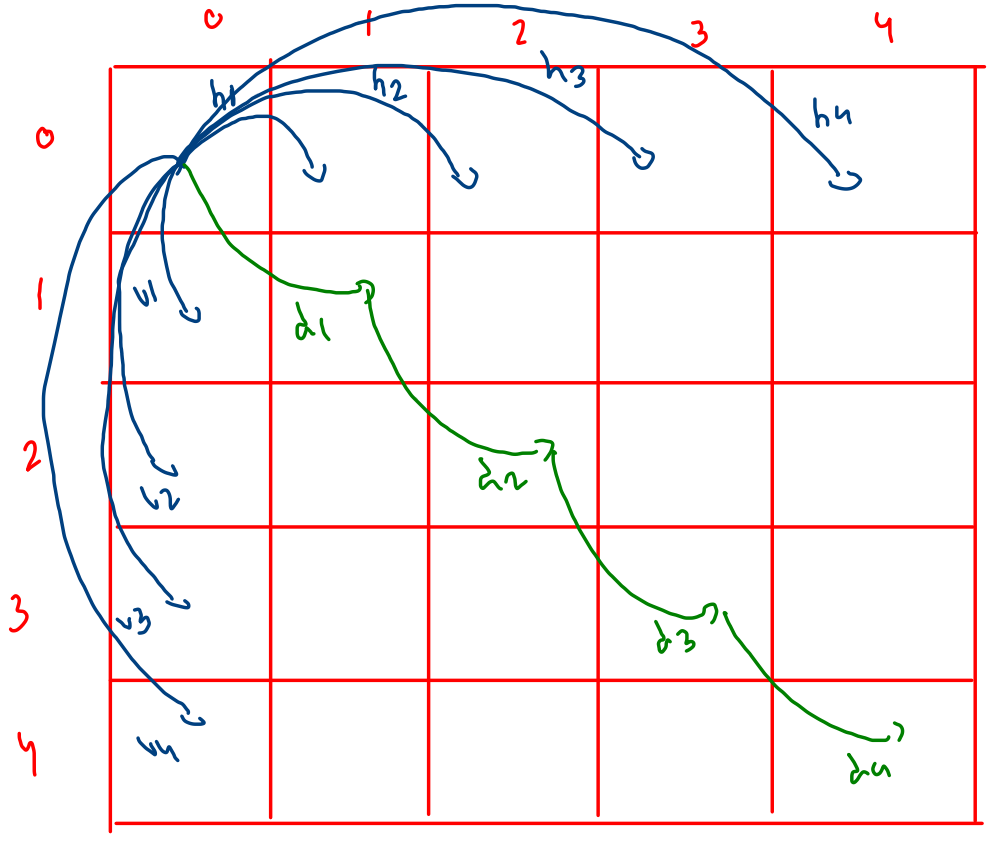


dr = 2

dc = 2

```java
if(sr == dr && sc == dc) {
    ArrayList<String>ba = new ArrayList<>();
    ba.add("");
    return ba;
}

if(sr > dr || sc > dc) {
    ArrayList<String>ba = new ArrayList<>();
    return ba;
}

ArrayList<String>hntd = getMazePaths(sr,sc+1,dr,dc); // horizontal nbr to dest paths
ArrayList<String>vntd = getMazePaths(sr+1,sc,dr,dc); // vertical nbr to dest paths

ArrayList<String>stod = new ArrayList<>(); // src to dest paths

//src to dest ->'h' + horizontal nbr to dest
for(int i=0;i < hntd.size();i++) {
    stod.add('h'+ hntd.get(i));
}

//src to dest -> 'v' + vertical nbr to dest
for(int i=0; i < vntd.size();i++) {
    stod.add('v' + vntd.get(i));
}

return stod;
```




0
1
2
3
4
0
1
2
3
4
h1
h2
h3
h4
v1
v2
v3
v4
d1
d2
d3
d4

```java
//horizontal nbr
for(int i=1; sc + i <= dc ; i++) {
    ArrayList<String>hntd = getMazePaths(sr,sc + i,dr,dc);

    for(int j=0; j < hntd.size();j++) {
        std.add("h" + i + hntd.get(j));
    }
}

//vertical nbr
for(int i=1; sr + i <= dr ; i++) {
    ArrayList<String>vntd = getMazePaths(sr + i,sc,dr,dc);

    for(int j=0; j < vntd.size();j++) {
        std.add("v" + i + vntd.get(j));
    }
}

//diagonal nbr
for(int i=1; sr + i <= dr && sc + i <= dc ; i++) {
    ArrayList<String>dntd = getMazePaths(sr + i ,sc + i,dr,dc);

    for(int j=0; j < dntd.size();j++) {
        std.add("d" + i + dntd.get(j));
    }
}
```