

Recursion → Normal Recursion

Post

↳ Array → Traversal } - All Indices
↳ ArrayList } Get → Dynamic Size
↳ Recursion with String

Importance
of pre & { Return
part = type

↳ Recursion on tree
↙ way up.

Bottom - to - Top }

i) final answer
→ Base case

Recursion Area

i) Pre Area

ii) In Area

iii) Post Area

→ Level i → Understanding / Learning.

→ Level j =

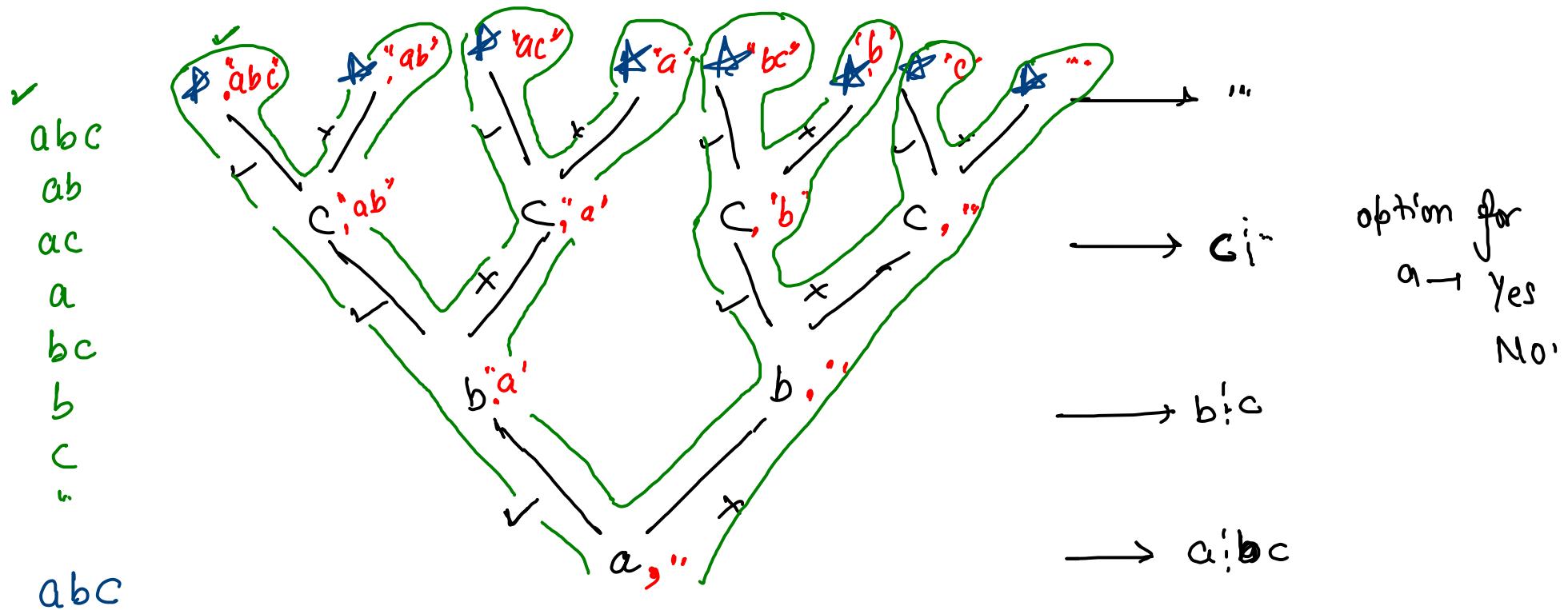
print Subsequence

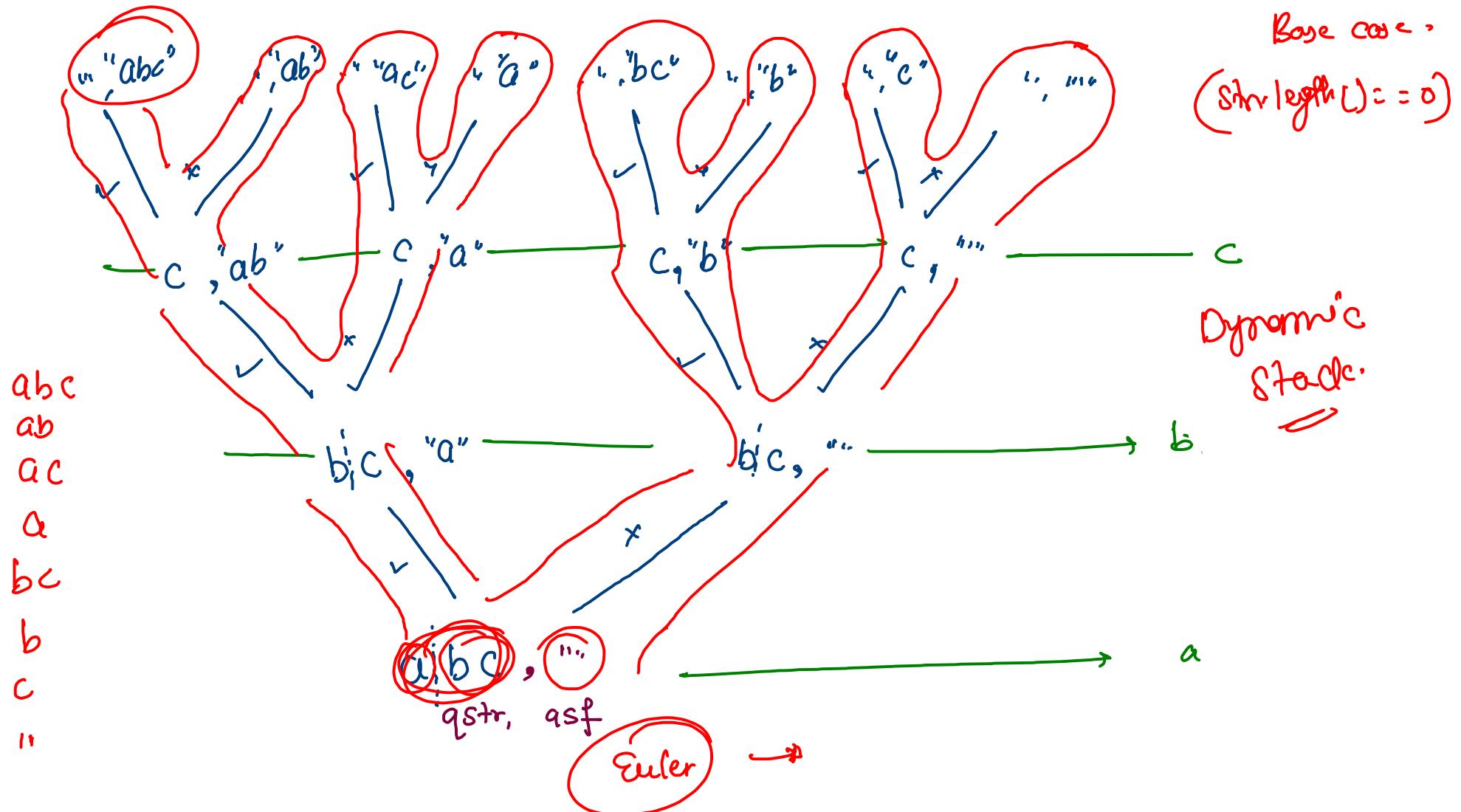
✓ levels

✓ options

level → base base-

options → No. of cells.





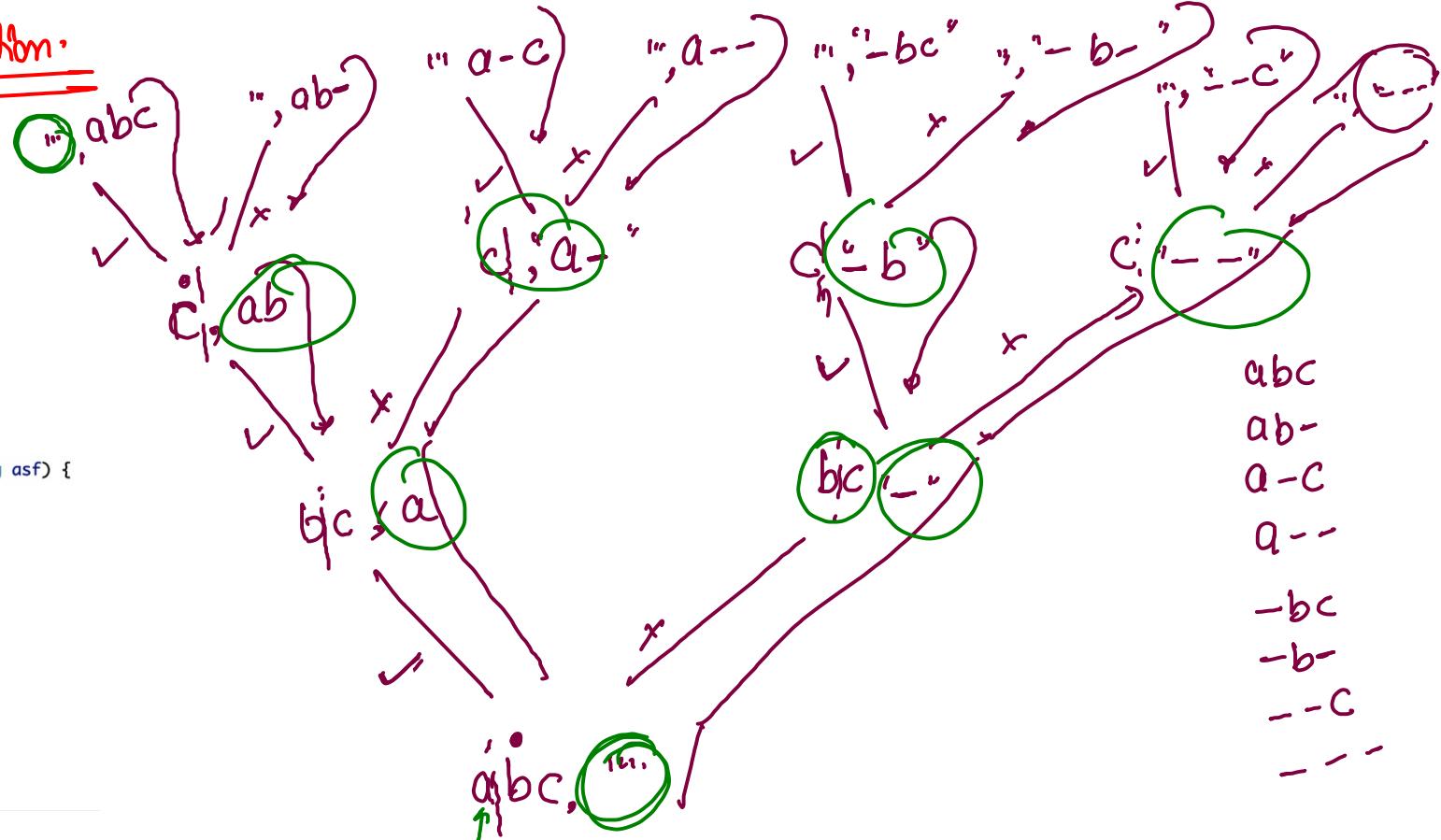
Expectation - ~~points $S(\underline{abc}, "")$~~ →
fact - ~~points $S(\underline{bc}, \underline{a})$~~ → ~~points $S(\underline{bc}, "")$~~ } ~~b1bc~~
} Analysis → perform

String → level & option:

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

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

    // yes call
    printSubseq(roq, asf + ch);
    // no call
    printSubseq(roq, asf + "-");
}
```

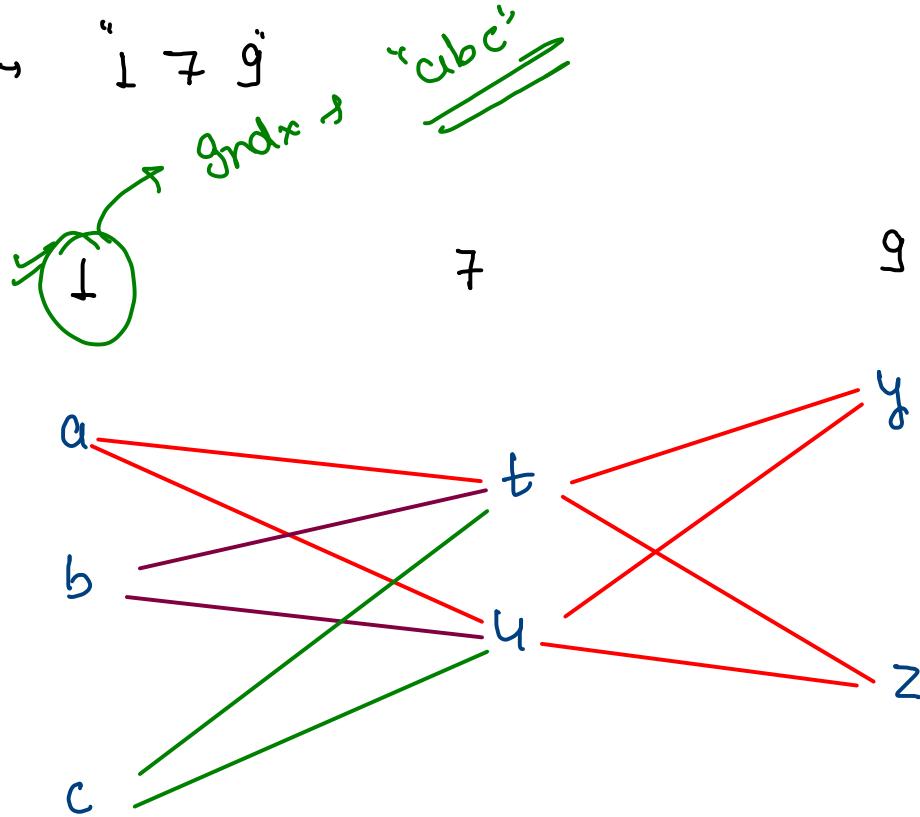


Print KPC

Key pad Code

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

ques + "1 7 9" "abc"



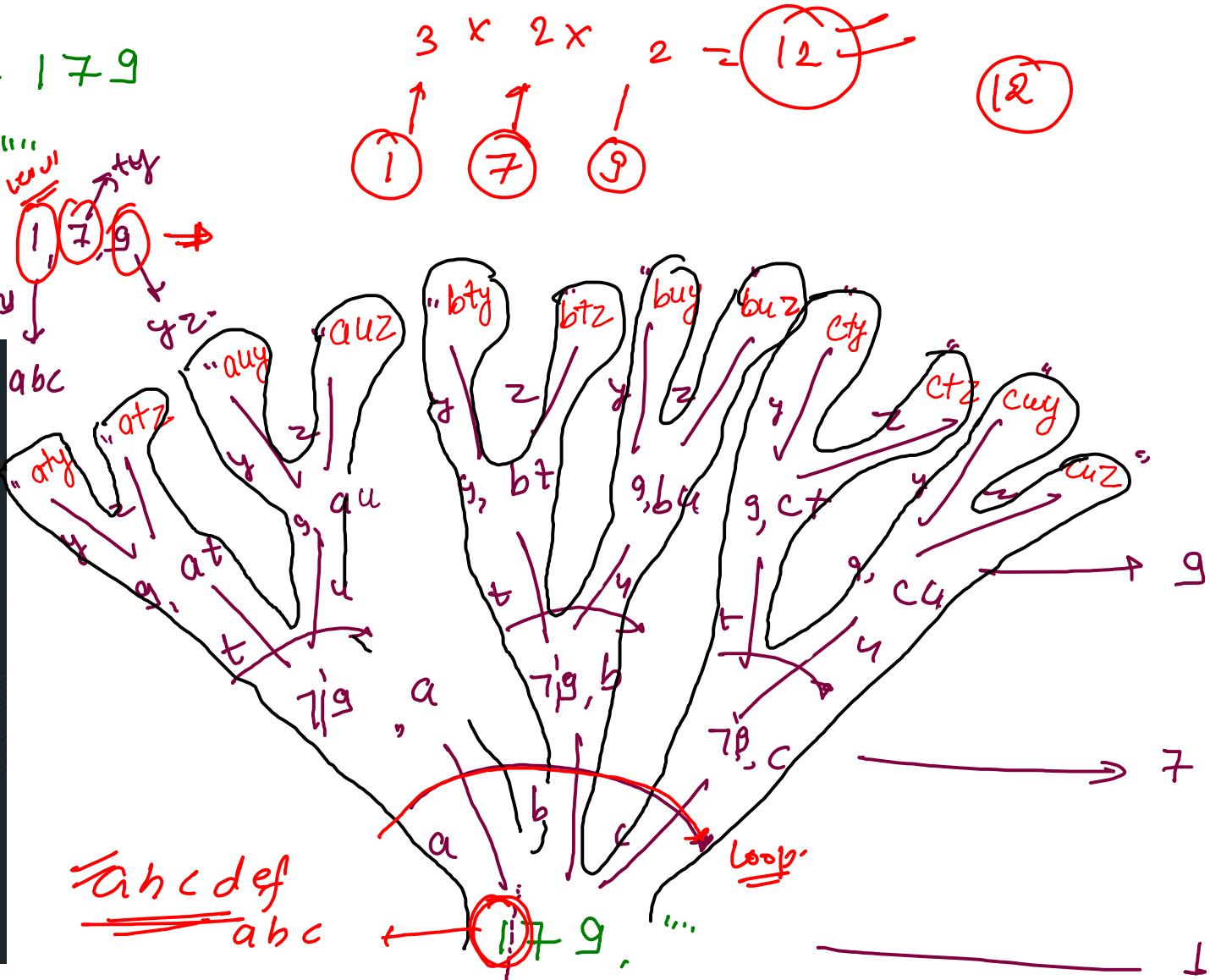
aty
atz
auy
auz
btg
btz
buy
buu
cty
ctz
cuy
cuu

qstr - 179

ans = " "
 " " "
 " " "

level → 1, 7, 9 →
options ↘

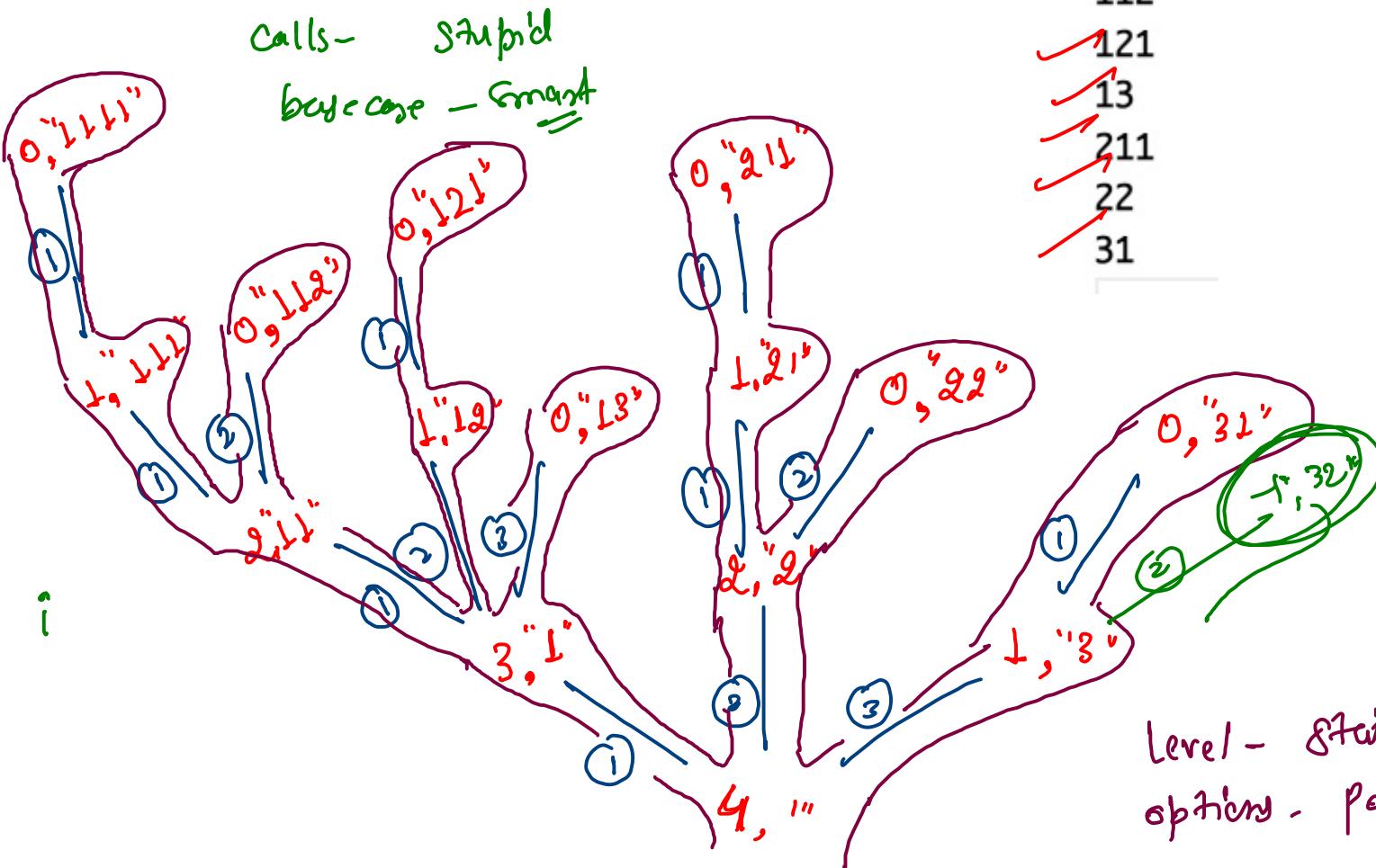
0 -> .;
1 -> abc
2 -> def
3 -> ghi
4 -> jkl
5 -> mno
6 -> pqrs
7 -> ty
8 -> vwx
9 -> yz



city
at_z
auy
942
b_ty
bt_z
buy
buz
cty
ct_z
cuy
gu_z

$n=4$

smart cells \rightarrow Invalid
Intermediate \times



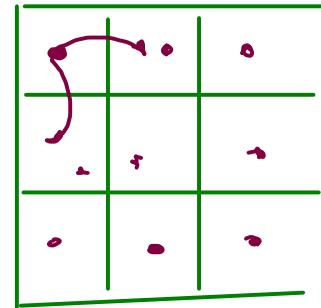
- ✓ 1111
- ✓ 112
- ✓ 121
- ✓ 13
- ✓ 211
- ✓ 22
- ✓ 31

level \rightarrow stairs.
option \rightarrow jumps from
Every stair.

- ✓ 1 1 1 1
- ✓ 1 2
- ✓ 2 1
- ✓ 1 3
- ✓ 2 1 1
- ✓ 2 2
- ✓ 3 1

Level - stair
options - possible jump -
No invalid jumps are allowed

print Marcpath.



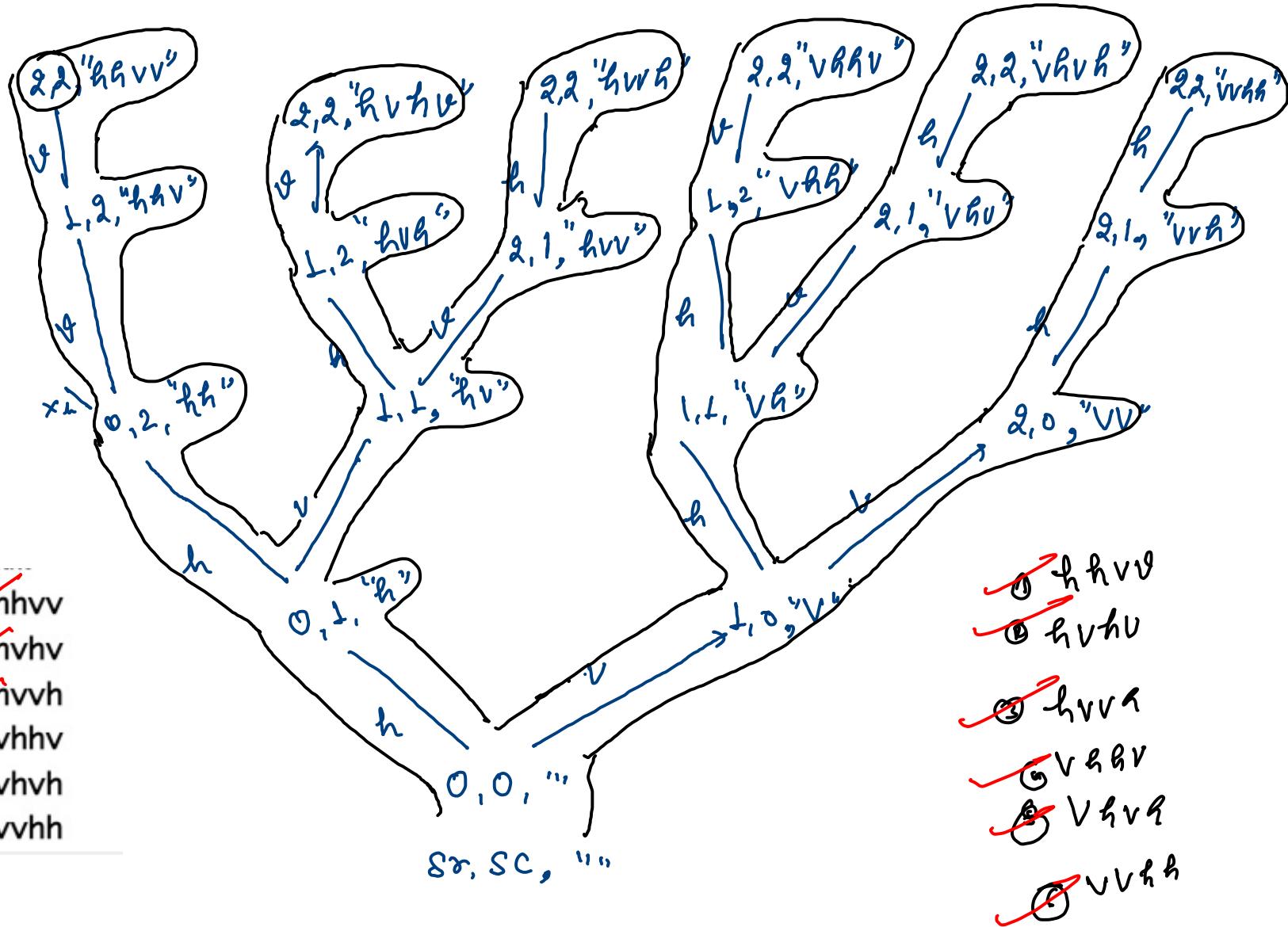
A diagram illustrating a transition between states. A horizontal red arrow points from the state $r_{i,c}$ to the state $r_{i,c+1}$. A vertical blue arrow points downwards from the state $r_{i,c}$ to the state $r_{i+1,c}$.

$$ds = \lambda$$

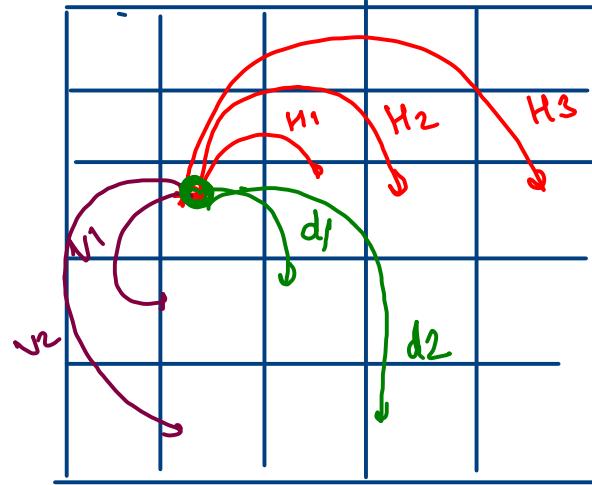
$$dc = 2$$

✓ hhvv
✓ hvhv
✓ hvhv
✓ vhhv
✓ vhvh
✓ vvhh

Smart - call



Print Maze path with jumps →



① ruler →
Horizontal
vertical
Diagonal

(r_i, c)
;
 r_{i+1}, c
 r_{i+2}, c
;
 r_{i+jump}, c
 $r_{i+jump} \{ dr$

$r, c+1, r, c+2, \dots, r, c + jump \leq dc$

$r_{i+jump}, c+jump$
 $i+jump \leq dr \& c+jump \leq dc$



