

Crossword Puzzle

A hand-drawn weather map of Europe on a grid. Key features include:

- Cities:** PARIS, ANKARA, LONDON.
- Fronts:** A red dashed line labeled 'Vertical' and 'Horizontal' with arrows pointing upwards and to the right respectively. Other red dashed lines are labeled 'N' (North) and 'S' (South).
- Pressure Systems:** Red circles labeled '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'. Some are enclosed in circles.
- Wind Symbols:** Small black '+' and '-' signs indicating wind direction.

- ① Perfectly fit in slot
- ② Sharing possible then it must be shared.
- ③ All the space must be used

Hint → level words

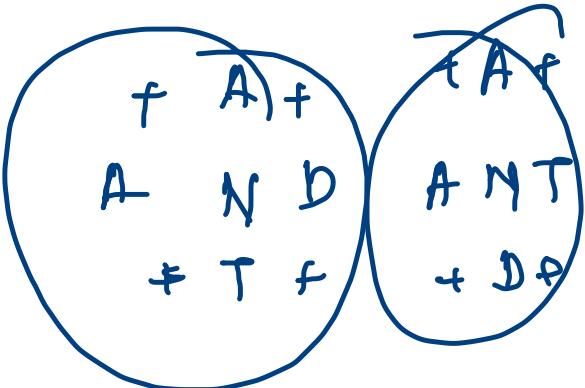
LONDON

→ Horizontally
→ vertically

option → cell of grid

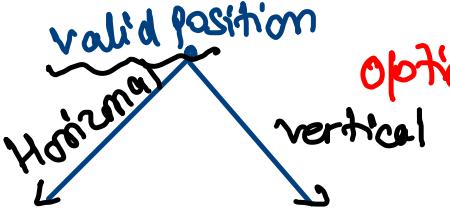
+	-	+
-	-	-
+	-	+

ANT }
AND }



Level → 0

LONDON



Level → 1

DELHI

Level → 2

ICELAND

Level → 3

ANKARA

RULE:

- ① Two horizontal words are not allowed. [words will fill perfectly in slot]
- ② Two vertical words are not allowed.
- ③ Horizontal and vertical cross allowed and share characters.

2D traversal
on grid

option → 2D matrix Traversal

Always begin with $i=0$ & $j=0$

because of all permutation

$\sim .11 \text{str.charAt}(0) == grid[i][j]$

① can I place word Horizontally?
Boolean place word Horizontally } preset
array return Run function.
call

③ unplace word Horizontally. } to
preser
④ can I place word vertically?
call

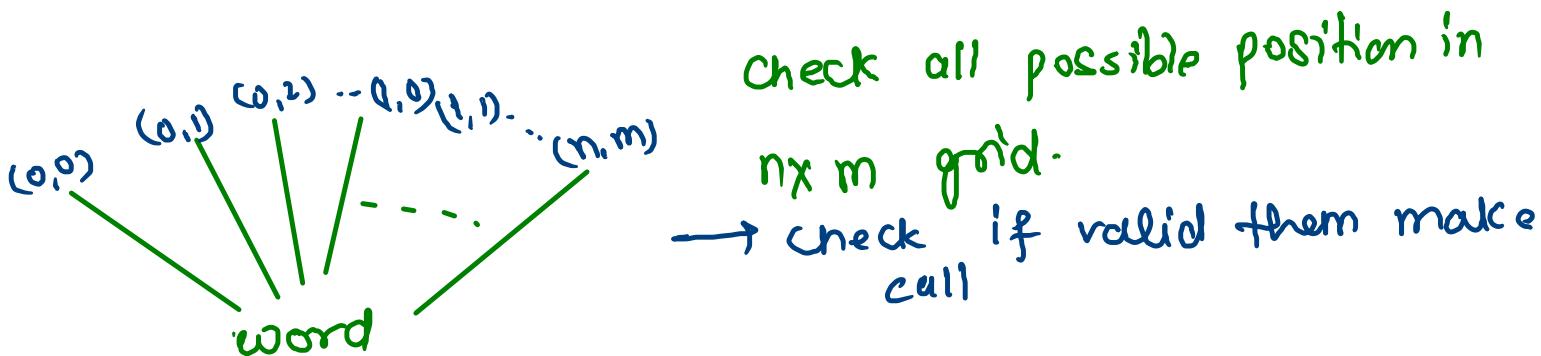
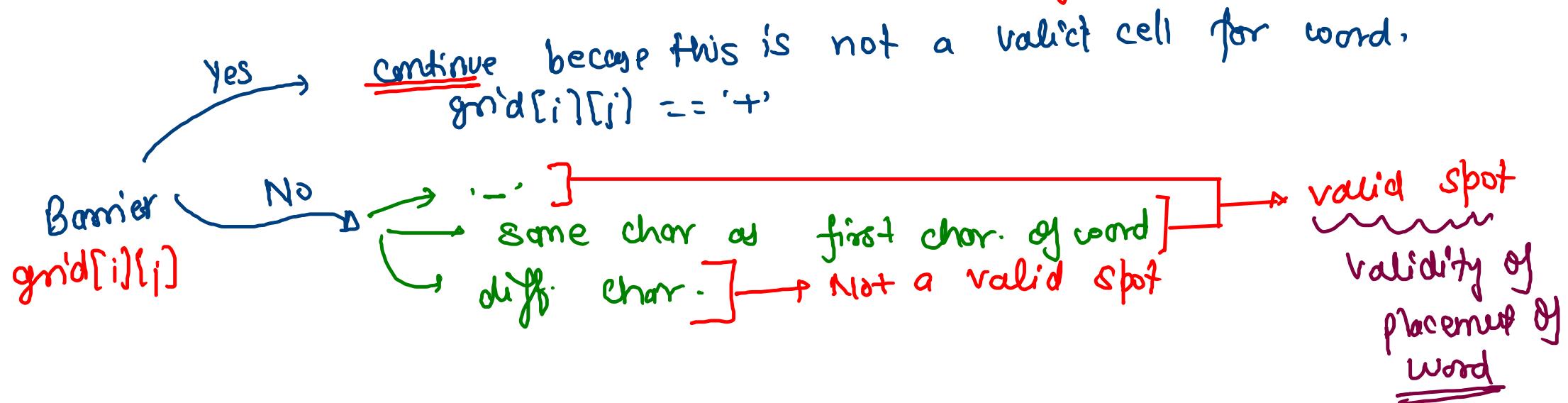
⑤ place word vertically.

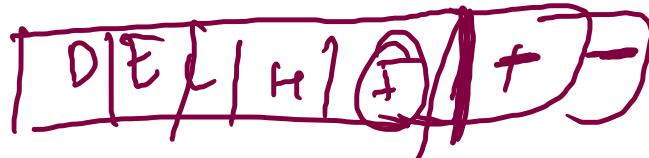
call

⑥ unplace word vertically.

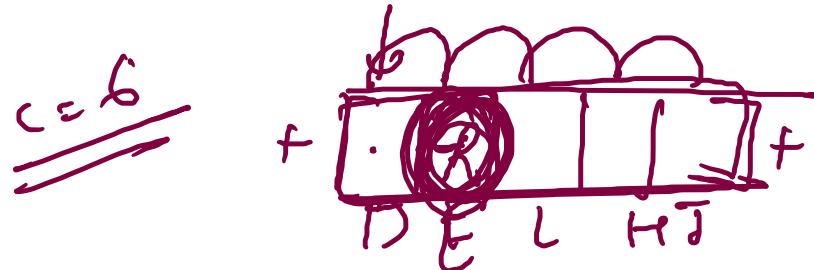
→ 6 functions are needed.

When we give chance to cell for checking of validity of word.





$\underline{1-1} + \underline{5} = \textcircled{S}$ T word.
leges



~~$c=0$~~
 $\underline{\underline{1}} + \underline{\underline{5}} = \textcircled{S}$ T word.

$\underline{\underline{0}} + \underline{\underline{5}} = \textcircled{S}$ T word.

```
// crossword
public static boolean canPlaceHorizontal(char[] grid, int r, int c, String word) {
    // left check
    if(c > 0 && grid[r][c - 1] != '+') {
        return false;
    }
    // right check
    if(c - 1 + word.length() >= grid[0].length) {
        return false;
    }
    if((c - 1 + word.length() < grid[0].length - 1) && (grid[r][c + word.length()] != '+')) {
        return false;
    }
    for(int j = 0; j < word.length(); j++) {
        if(grid[r][j + c] != '-' && grid[r][j + c] != word.charAt(j)) {
            return false;
        }
    }
    return true;
}
```

How to prepare
slots

	0	1	2	3	4	5	6	7	8	9
0	+	-	+	+	+	+	+	+	+	+
1	+	-	+	+	+	+	+	+	+	+
2	+	-	+	+	+	+	+	+	+	+
3	+	-	-	-	-	+	+	+	+	+
4	+	-	+	+	+	-	+	+	+	+
5	+	-	+	+	+	-	+	+	+	+
6	+	+	+	+	+	-	+	+	+	+
7	+	+	-	-	-	-	-	-	-	+
8	+	+	+	+	+	-	+	+	+	+
9	+	+	+	+	+	-	+	+	+	+

vertical slots

Starting point \rightarrow x,y

length of slot

$\rightarrow (0,1,6), (3,5,7)$

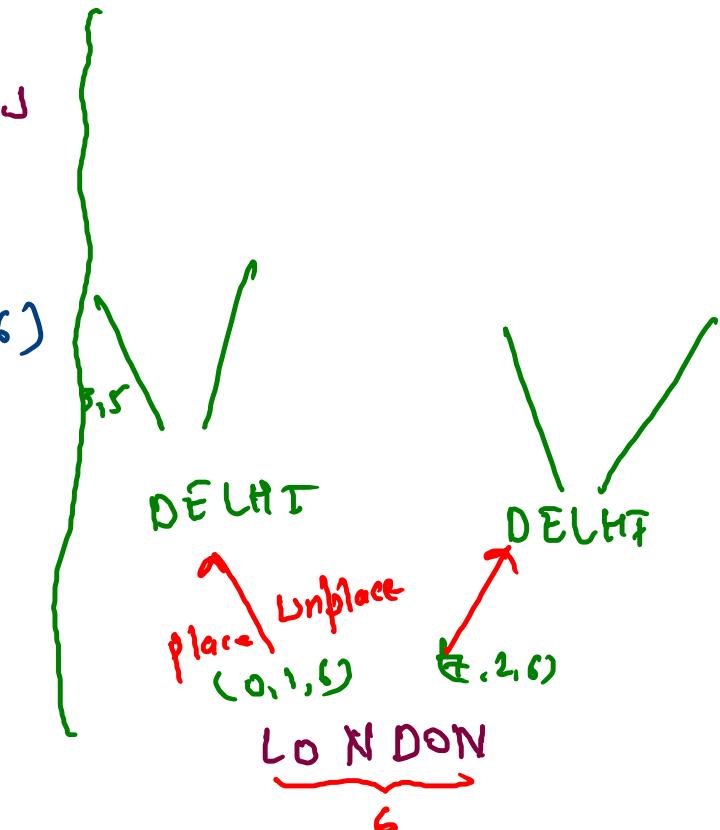
Horizontal slots

Starting point, x,y

length of slot

$\rightarrow (3,1,5), (7,2,6)$

④ LONDON, DELHI,
ANKARA, ICELAND



K-partitions

n = length of array

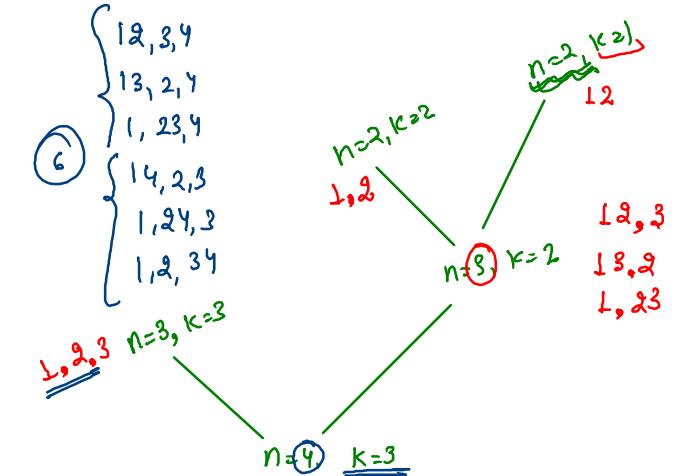
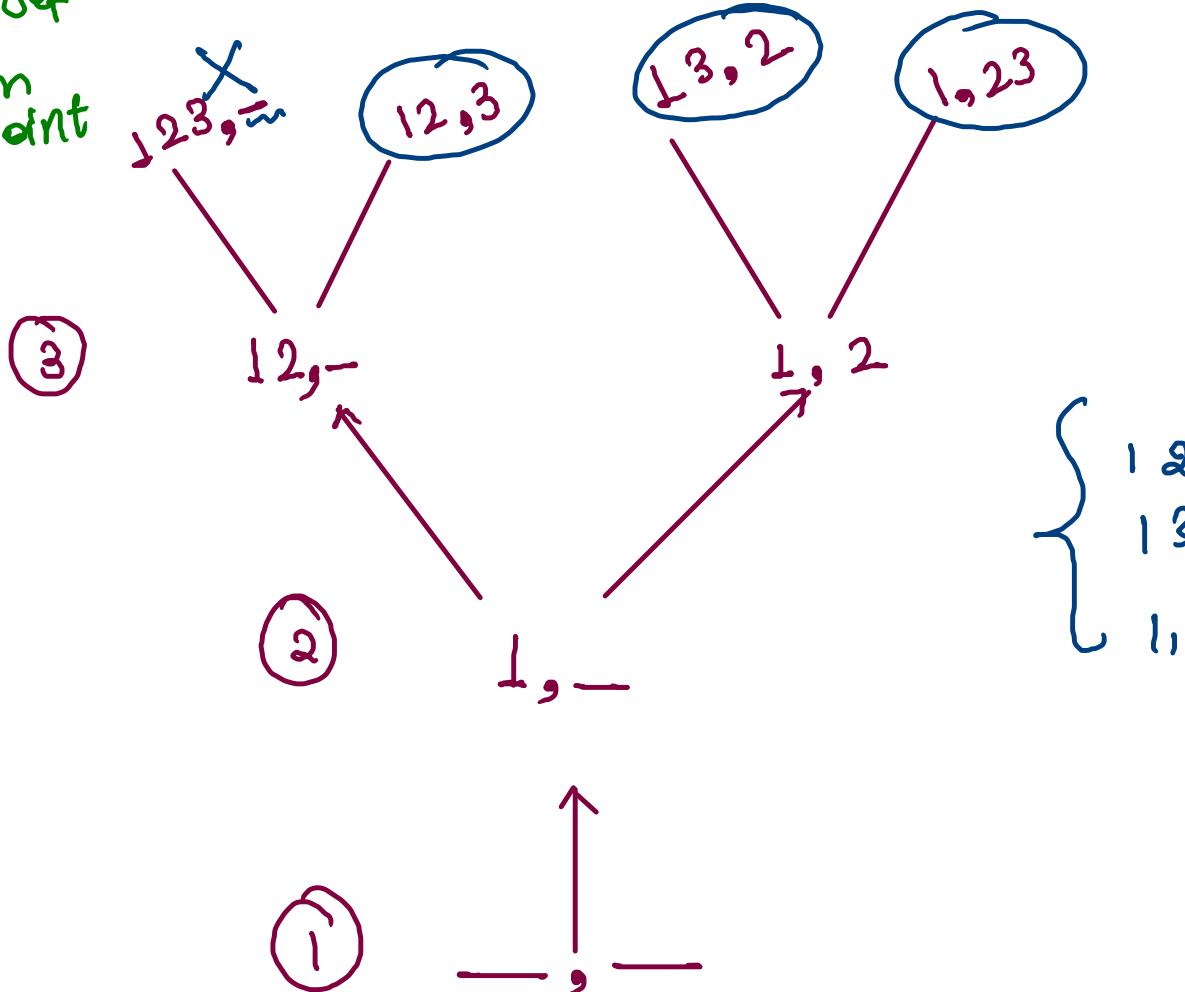
$k \rightarrow$ No. of partition Required

level \rightarrow 1

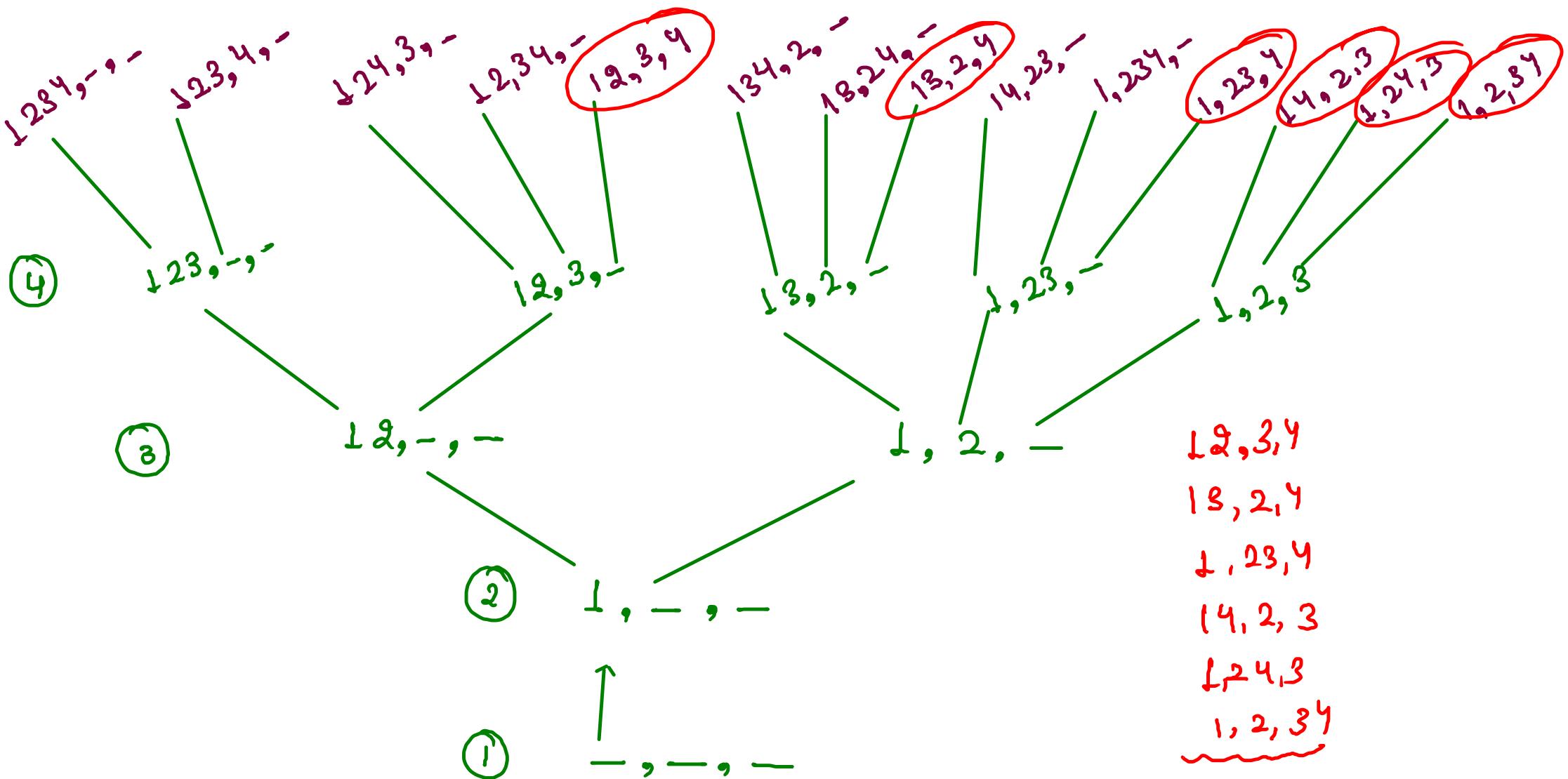
option \rightarrow Existing Set

Start from
current point

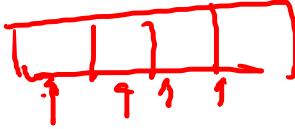
$n=3$
 $k=2$



$n=4$
 $k=3$



Implementation based discuss.



For i =
No... < n)

