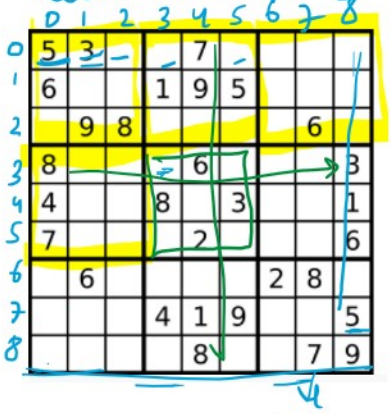
24th Feb

24 February 2022 21:13

https://leetcode.com/problems/valid-sudoku/

Valid Sudoku :-



9 × 9

1. Each row must contain the digits `1-9` without repetition.
2. Each column must contain the digits `1-9` without repetition.
3. Each of the nine `3 x 3` sub-boxes of the grid must contain the digits `1-9` without repetition.

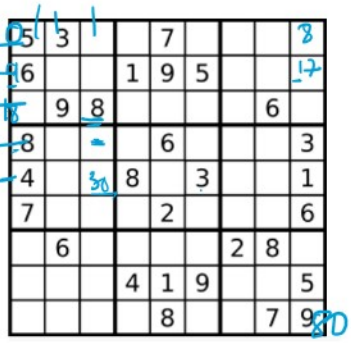
- Filled -
- Not Filled - Ignore

r c
7, 8

r - (r % 3)
c - (c % 3)

7 - (7 % 3) = 6
8 - (8 % 3) = 6

Total no. of elements = 81



20

r = 20 / 9 = 2,]-2,2

c = 20 % 9 = 2

38

r = 38 / 9 = 4

c = 38 % 9 = 2

```
boolean vS(sudoku, n){
  if(n==81) return true;
  r = n/9;
  c = n%9;
  if(sudoku[r][c] != '.'){
    if(!isValidElement(sudoku, r, c))
      return false;
  }
```

}

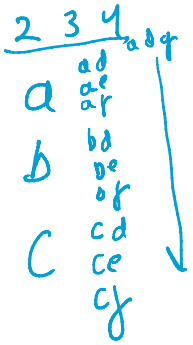
return vS(sudoku, n+1);

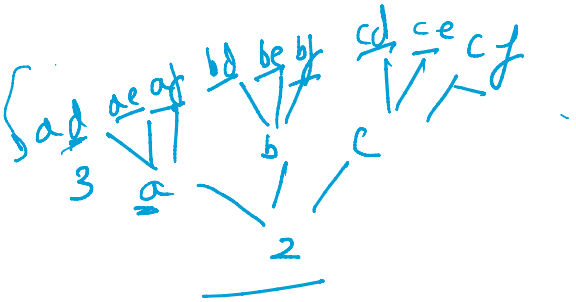
}

Keypad Combinations :- https://leetcode.com/problems/letter-combinations-of-a-phone-number

0 - ("")
1 - ("abc")
2 - ("def")
⋮

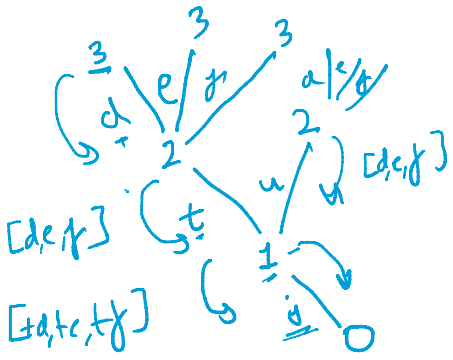
'23'
'0' = 31
'2' = 33
'2' - '0' = 33 - 31 = 2






```java
public static ArrayList<String> combination(String s, ArrayList<String> keypad, int i) {
    if (i == s.length()) {
        ArrayList<String> res = new ArrayList<>();
        res.add("");
        return res;
    }
    int pressedButton = s.charAt(i) - '0' - 1;
    String pressedString = keypad.get(pressedButton);
    ArrayList<String> res = new ArrayList<>();
    for (int j = 0; j < pressedString.length(); j++) {
        ArrayList<String> curr = combination(s, keypad, i + 1);
        for (int k = 0; k < curr.size(); k++) {
            res.add(pressedString.charAt(j) + curr.get(k));
        }
    }
    return res;
}
```

[g+d, j+e, j+f]
[+d, +e, +f]



S = "583"
i = 0