

Lexicographical Numbers : Lexicographical order \rightarrow Dictionary order

$a \rightarrow$	$1 \rightarrow$
$b \rightarrow$	$2 \rightarrow$
$c \rightarrow$	$3 \rightarrow$
$d \rightarrow$	$4 \rightarrow$
$da \rightarrow$	$4_1 -$
$db \rightarrow$	$4_2 -$
$dc \rightarrow$	$4_3 -$
$dca \rightarrow$	4_5
$dcb \rightarrow$	\vdots

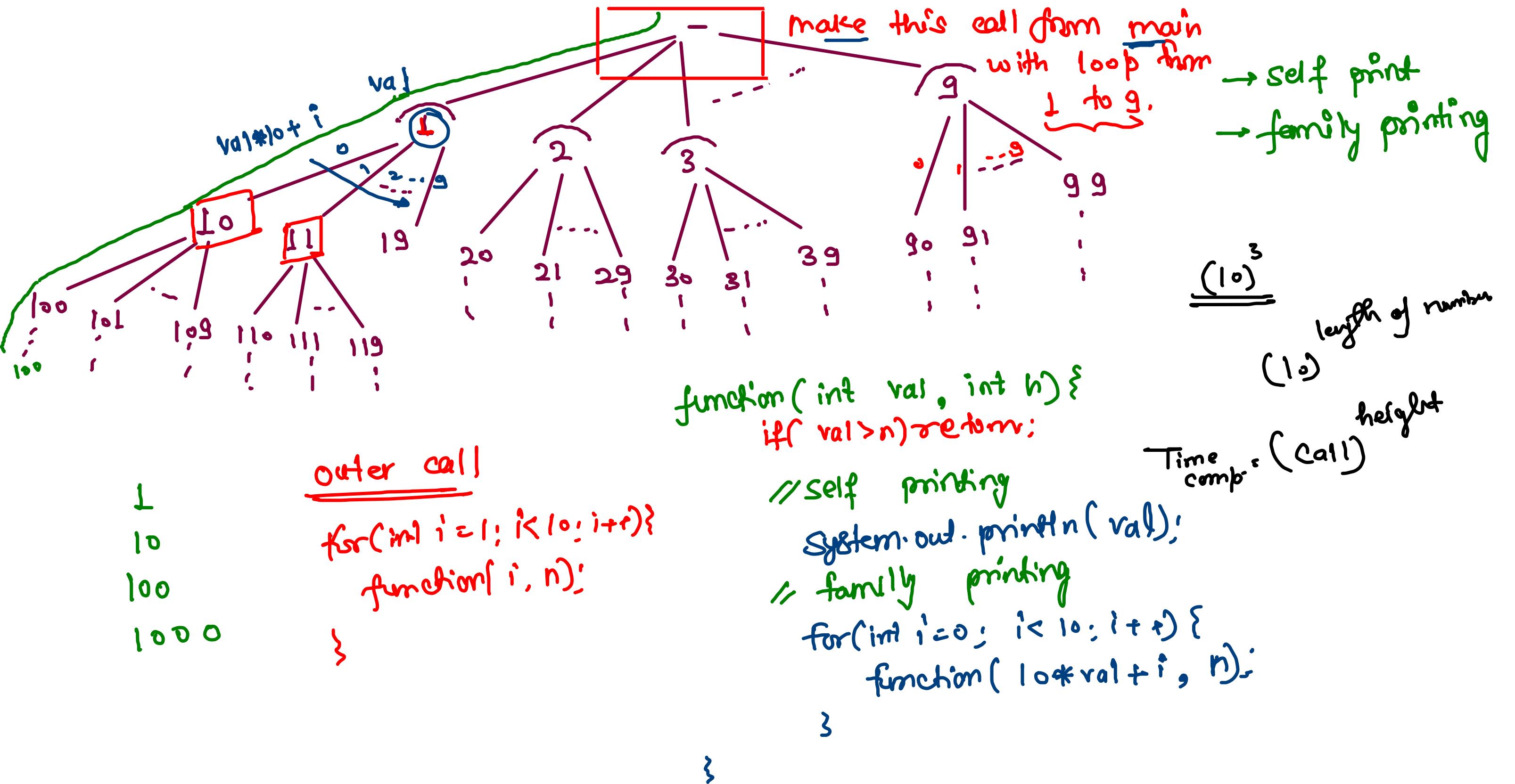
<u>n=1000</u>	print numbers in lexicographical order
1	1
10	11
100	12
1000	13
101	--
102	-
103	-
104	-
105	-
106	-
107	-
108	-
109	-
110	110
111	111
112	112
113	113
114	114
115	115
116	116
117	117
118	118
119	119
120	120
121	121
122	122
123	123
124	124
125	125
126	126
127	127
128	128
129	129
130	130
131	131
132	132
133	133
134	134
135	135
136	136
137	137
138	138
139	139
140	140
141	141
142	142
143	143
144	144
145	145
146	146
147	147
148	148
149	149
150	150
151	151
152	152
153	153
154	154
155	155
156	156
157	157
158	158
159	159
160	160
161	161
162	162
163	163
164	164
165	165
166	166
167	167
168	168
169	169
170	170
171	171
172	172
173	173
174	174
175	175
176	176
177	177
178	178
179	179
180	180
181	181
182	182
183	183
184	184
185	185
186	186
187	187
188	188
189	189
190	190
191	191
192	192
193	193
194	194
195	195
196	196
197	197
198	198
199	199
200	200
201	201
202	202
203	203
204	204
205	205
206	206
207	207
208	208
209	209
210	210
211	211
212	212
213	213
214	214
215	215
216	216
217	217
218	218
219	219

29
290
291
. .
;
;
;
299

3
30
300
301
302
. .
309

9
90
900
901
902
. .
909

91
910
911
912
. .
. .



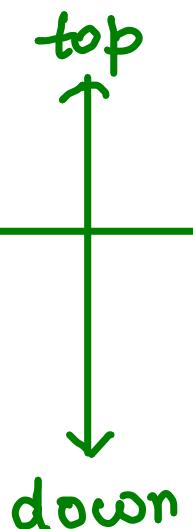
Gold Mine - 2

mine.

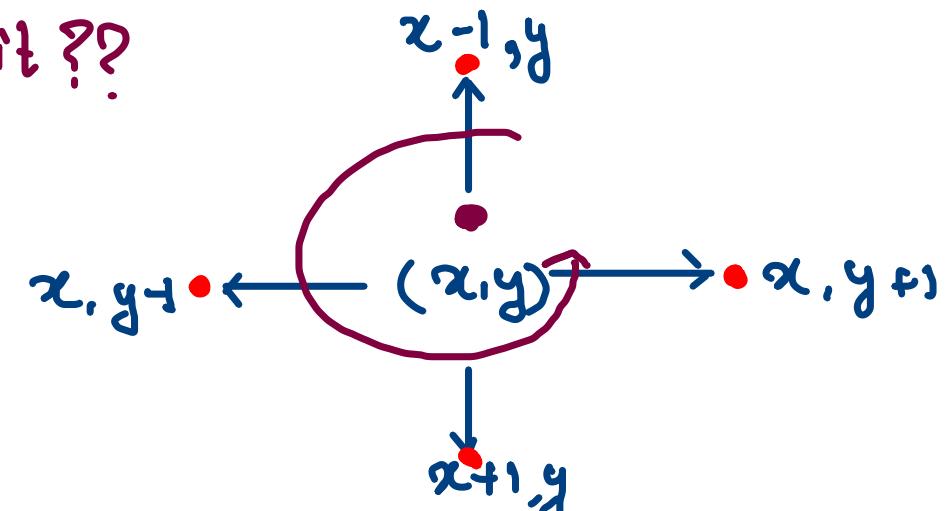
10	0	100	200	0	8	0
20	0	0	0	0	6	0
30	0	0	9	12	3	4
40	0	2	5	8	3	11
0	0	0	0	0	9	2
5	6	7	0	7	4	2
8	9	10	0	1	10	8

$\text{mine}[i][j] == 0$ Barrier

we can't mining
Here



Max · Profit ??



Solve Sudoku

$\text{arr}[i][j] == 0$

Search Row 1 to 9 → Row ↓

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

is Safe.

safety

Each col ↓
1 to 9

Each submatrix
1 to 9

4,5

4,6

4-4/3

5-5/3

$$m = \left\lfloor \frac{r - 1}{3} \right\rfloor$$

$$c = \left\lfloor \frac{c - 1}{3} \right\rfloor$$

Similar

starting point of Submatrix →

```
for(int j=0; j<n; j++) {
    if(arr[row][j]==n)
        return false;
}
```

$$rr = r - r \% 3 \quad \{$$

$$cc = c - c \% 3 \quad \}$$

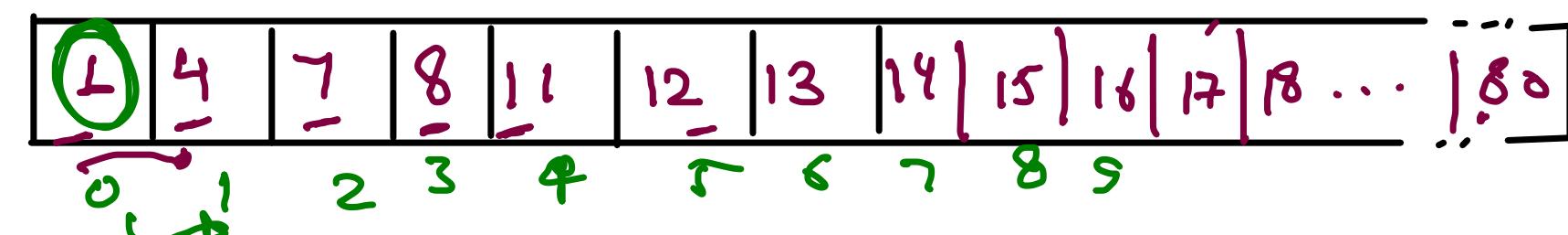
Recursive function

```
if(arr[i][j]==0) {
    for(int n=1; n<10; n++) {
        if(isSafe(arr, i, j, n)) {
            arr[i][j] = n;
            if(solveSudoku(arr))
                return true;
            arr[i][j] = 0;
        }
    }
}
```

Method to solve Sudoku with 1D array.

① Fill box Number of Empty cell in
a one-D array.

Array -



herr + c

$g \times 1$ f 4 $\text{Gndx} \rightarrow$ array(of b₁₀ Num)

13

$$O \rightarrow \begin{cases} \text{total O}_2 \\ \text{total N}_2 \end{cases}$$

Cryptarithmetic :- String S_1 , string S_2 , string S_3 . find out mapping of character, such that $S_1 + S_2 = S_3$

$S_1 \rightarrow SEND$
 $S_2 \rightarrow MORE$
 $S_3 \rightarrow MONEY$

$$\begin{array}{r} \cancel{S} E N D \\ \cancel{M} O R E \\ \hline \cancel{M} O N E Y \end{array} \quad \begin{array}{r} 9 S 6 7 \\ - 1 0 8 5 \\ \hline 1 0 6 5 2 \end{array}$$

1 mapping from all possible,

$S \rightarrow 9$
 $E \rightarrow 5$
 $N \rightarrow 6$
 $D \rightarrow 7$
 $M \rightarrow 1$
 $O \rightarrow 0$
 $R \rightarrow 8$
 $Y \rightarrow 2$

...
 $9:18 \rightarrow 9:30$
 12 min.
 mapping - H.M
 S_1, S_2, S_3
 usedNumber
 uniqueString

Unique mapping

char \rightarrow

unique string generate.
 level \rightarrow character of string
 option \rightarrow 0 to 9

ASCII
 $V_1: S_1 \rightarrow [9nt]$
 $V_2: S_2 \rightarrow 9nt$
 $V_3: S_3 \rightarrow Snt$
 $V_1 + V_2 = V_3$
 mapping
 is const

S_1
 S_3

$s_1 \rightarrow SEND$, $s_2 \rightarrow MORE$ $s_3 \rightarrow MONEY$

UNIQUE \rightarrow

SEND MORY

level \rightarrow character of
string

option \rightarrow Digit from
0 to 9

Y
R
O
M

$n_1 \rightarrow \text{from } s_1$
 $n_2 \rightarrow \text{from } s_2$
 $n_3 \rightarrow \text{from } s_3$

$n_1 + n_2 = n_3$
Mapping is
correct

→ point snapping in
sorted order.

