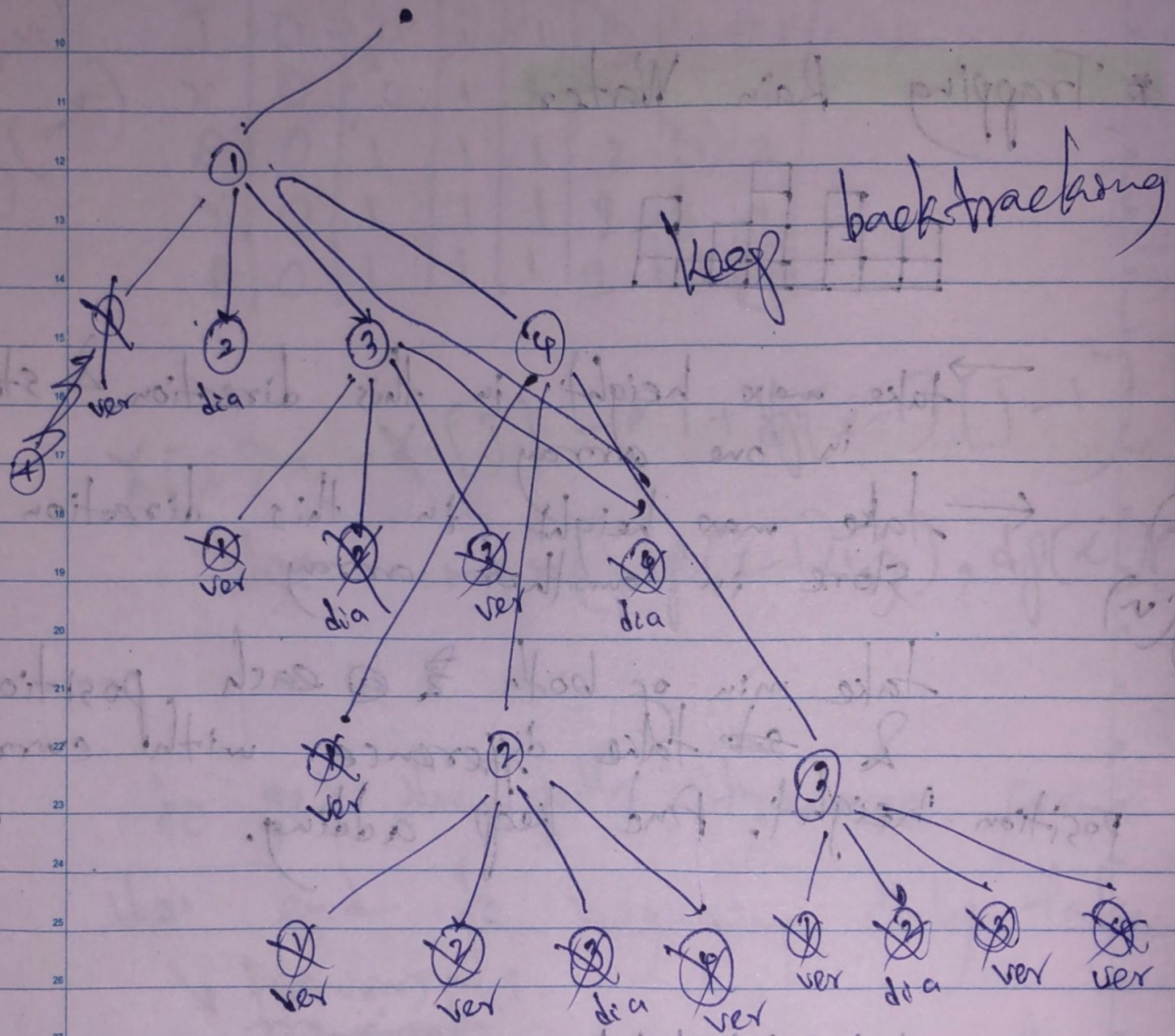
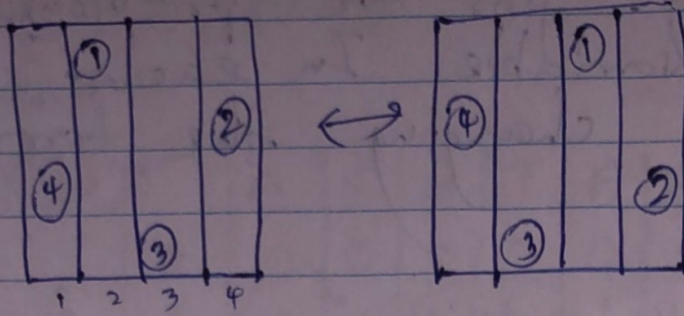


* N-Queens



Ignore

$O(n!)$
 $O(n^2)$

In each col, store queens place
Then check for validity over all
queens when storing in next
iteration.

	0	1	2	3
0	0	-1	-2	-3
1	1	0	-1	-2
2	2	1	0	-1
3	3	2	1	0

0 to 3 values

diagonal

col-row

	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6

0 to 6 values

col+row

Instead of storing in an array of each queen's place, we can use bit manipulation.

row, diagonal, \ diagonal variables

check if there's no queen crossing
↳ backtrack

(at the beginning n options. Then $(n-1)$ options.
... 2, 1, ... $\dots O(n!)$
upper bound $O(n^n)$