51 N-Queens

ans = [ ]

let answer[x] be col on which the x^{th} row queen is placed

we will places Queens recursively row by row

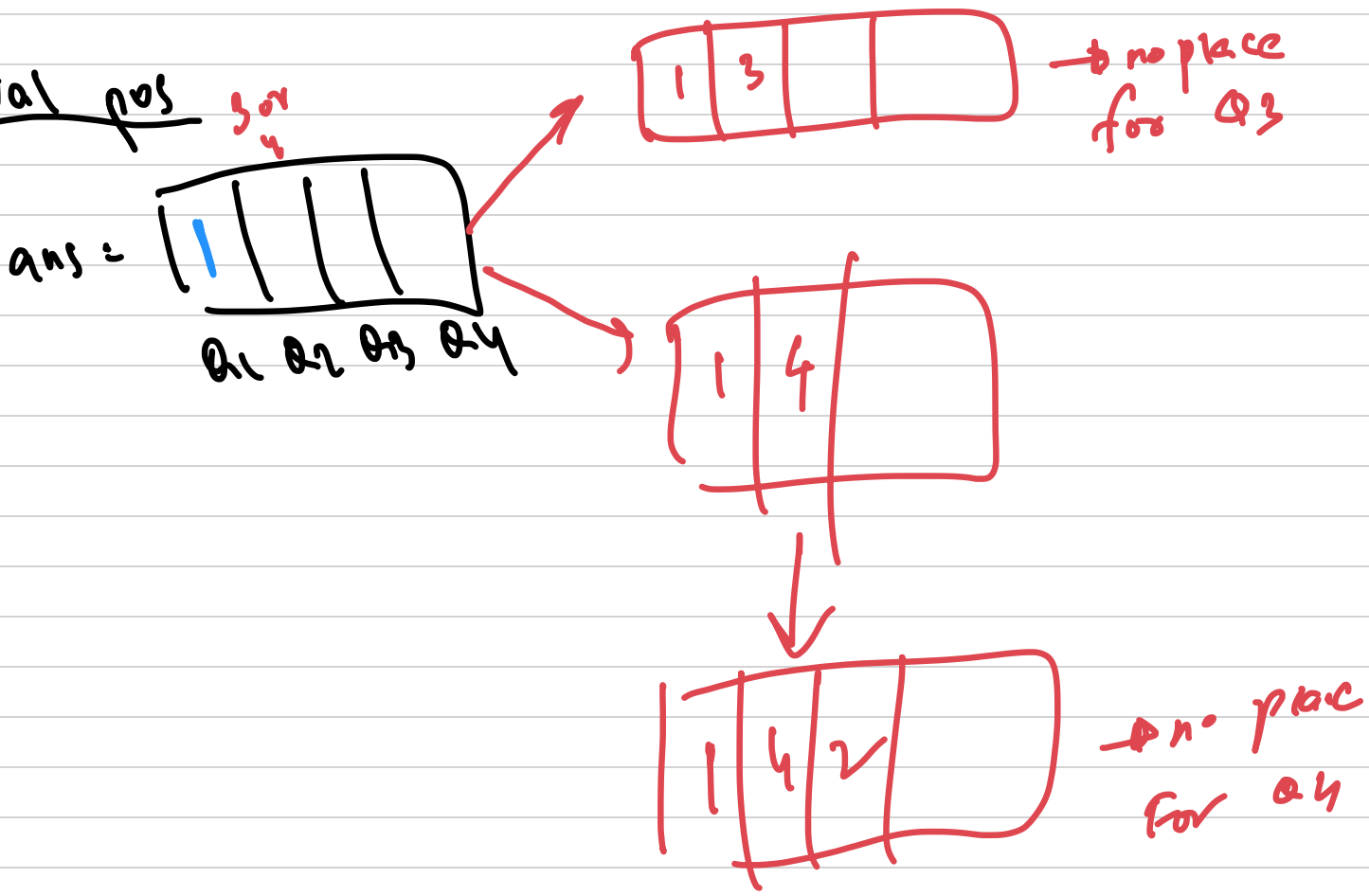
aval_pos = [[ ], [ ], [ ], ..]

aval_pos[x] contains all positions on which Queen of row x can be placed

ex say n=4

[(1,2,3,4) ~~(1,2,3,4)~~ ~~(1,2,3,4)~~ ~~(1,2,3,4)~~]

aval pos for 4



basic recursion idea

```
fnc upd_pos ()
{
  update all aval_pos of other
    Queens after inserting this
      Queen
```

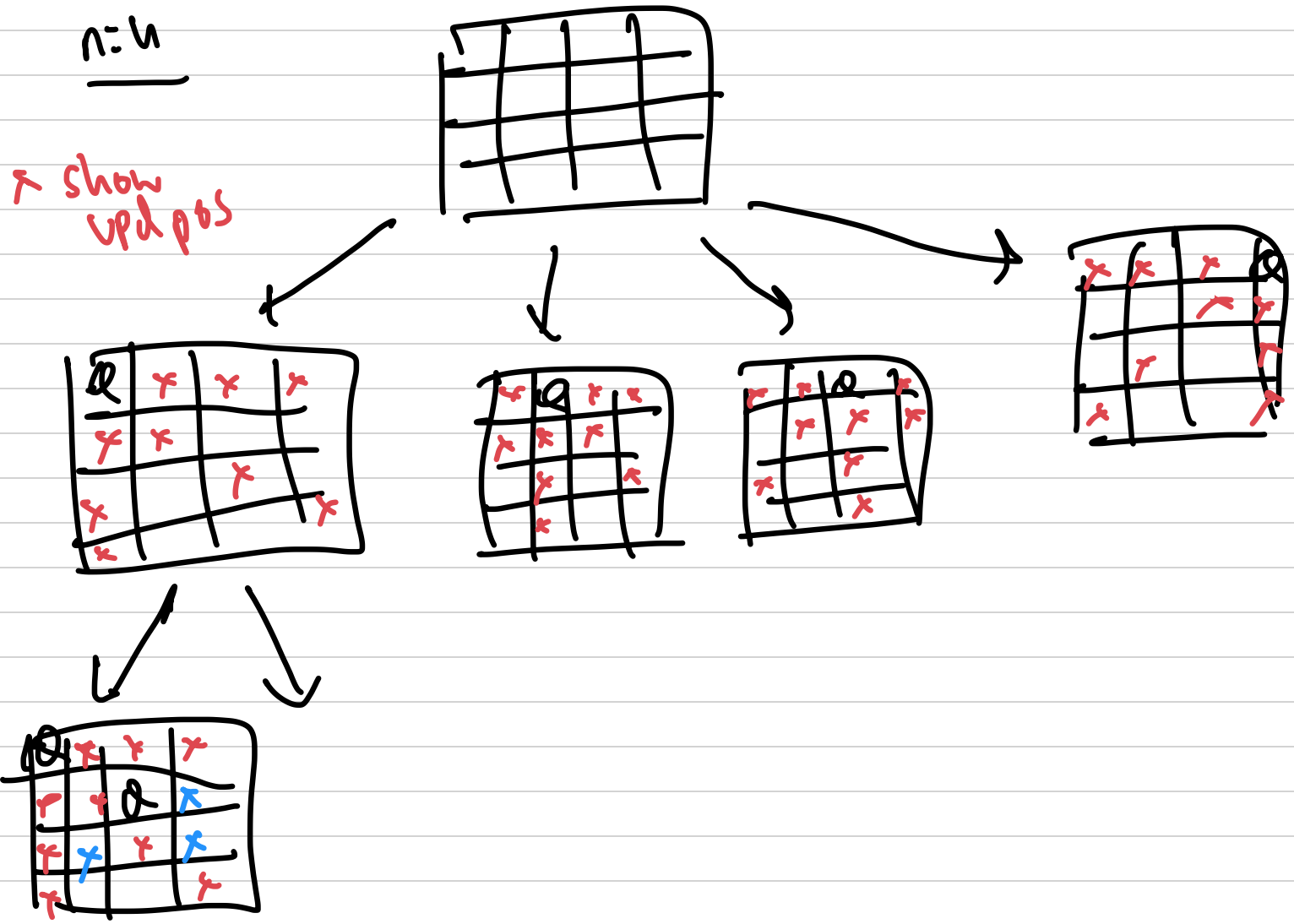
final_ans

```
fnc solve ( ans, curr_row, aval_pos )
if curr == n:
    final_ans.append(ans)
for c in avalpos[curr_row]:
    new_ans = copy.copy(ans)
    new_ans[curr_row] = avalpos[c]
    update avalpos
    self.solve(ans, curr_row+1, )
```

example

n: 4

↑ show upd pos



↓

stop

to place

for Q3

Similary all

branches expanded

if any found appended

in global ans array.