

COP 3502 - 12/1/2021

Backtracking

- Brute Force optimization

brute force - enumerate all possibilities,
See which ones "work".

(1) Divisibility

(2) 8-Queens

(3) Sudoku Solver

n digit number is k-divisible if for each
prefix of k digits, it's divisible by k.

5 digit : 22520

2 is divisible 1

22 is divisible 2

225 is divisible 3

2252 is divisible 4

22520 is divisible 5

BF try all 10^5 5 digit #s

13---

} these 1000 possibilities
aren't worth trying
since $13 \equiv 1 \pmod{2}$

In backtracking, we skip trying individual values in slots when they are "doomed to fail"

```
for (int i = 0; i < numChoices; i++) {  
    if (used[i]) continue;  
    if (doomedToFail(i, k, otherstuff)) continue;  
    // if placing i in slot k is doomed to fail
```

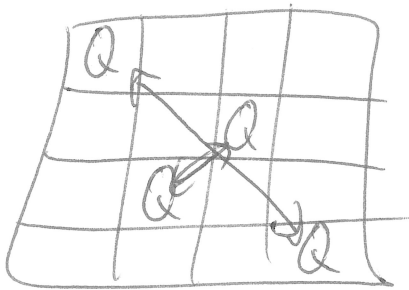


Runtime analysis theoretically is
VERY DIFFICULT for backtracking
solns.

8 QUEENS

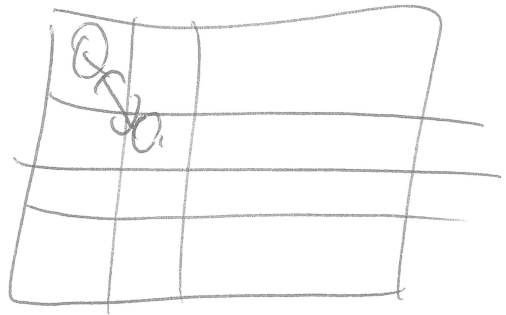
Place all 8 queens on the board
so that no two can attack
each other!

1, 3, 2, 4



bad

1, 2,



b.f. try 6! perms
starting 1, 2

backtracking - skip
placing the 2nd queen
in column 2!

SUDOKU SOLVER CODED LIVE

IDEA: In slot k , skip over
any values that cause an
inconsistency!