Backtracking 4/22/22

- optimization to brute force

Brute Force Examples

(1) Permutations
(2) Combination
(3) Or another arrangement of all possibilities

- list each option
- evaluate each option

Example NOT all Perms Possible

Visit each location once,
but not all pairs of ~~roads~~ locations
are connected

n=10 no road 3 → 5

3, 5, -, -, -, -, -, -, -, -] doomed to fail

↳ 8! wasted work

Instead of building all permutations,
skip prefixes that "doomed to fail".

In code

```
for (int i = 0; i < n; i++) {
    if (used[i]) continue;
    if (placing i in loc k is doomed to fail)
        continue;
    used[i] = 1;
    perm[k] = i;
    // recursion
    used[i] = 0;
}
```

BACKTRACKING

Build subset of toppings via bitwise ops

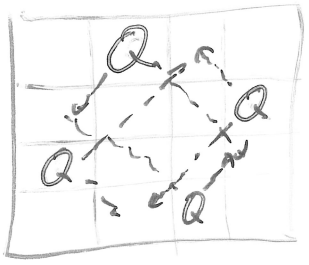
= 00000101 (topping 0, 2)

| 00001000] add topping 3)

→ Storage!

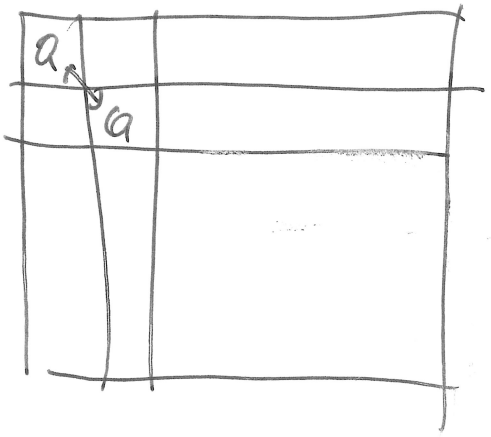
8 Queens

On a chessboard place 8 queens so no two attack each other.



row 0 col 1
row 1 col 3
row 2 col 0
row 3 col 2

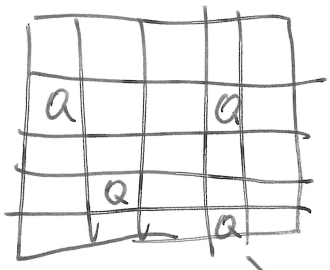
brute force try all perm of (0,1,2,3)
place queen on row i, col perm[i]
(1,3,0,2)



(0,1,-,-,-,-,-,-)
6! wasted work

Idea: Skip placing 1 in slot 1, since it's doomed to fail.

before placing i in slot k, we check if it conflicts with any previously placed queen.



(1,3)

(3,1)

$$\text{slope} = \frac{3-1}{1-3} = -1 \checkmark$$

$$abs(\partial x) == abs(\partial y)$$

(1,0) (4,3) $\frac{3-0}{4-1} = 1 \checkmark$

Digit Divisibility

k digit divisible if each of its
i digit prefixes is ~~d~~ divisible by i

6 digit divisible # is

126402

1 is divis 1
12 is divis 2
126 is divis 3
1264 is div 4
12640 is div 5
126402 is div 6

13 → nothing can start w/13.