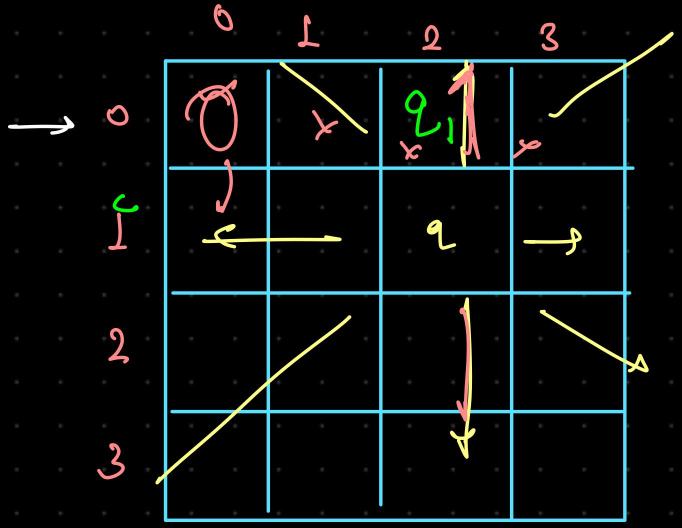
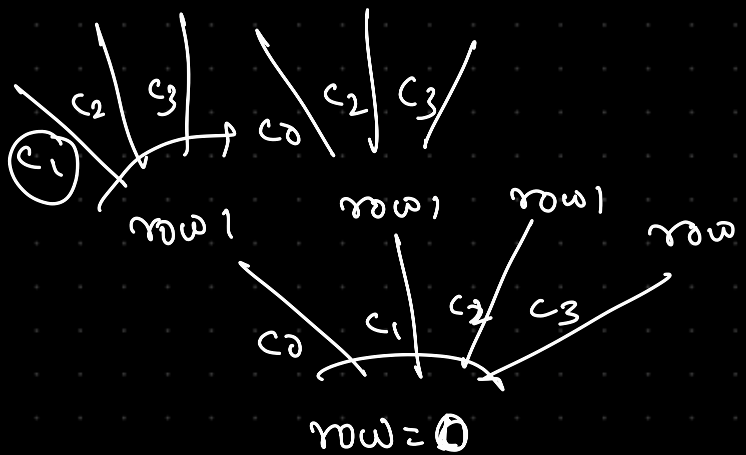
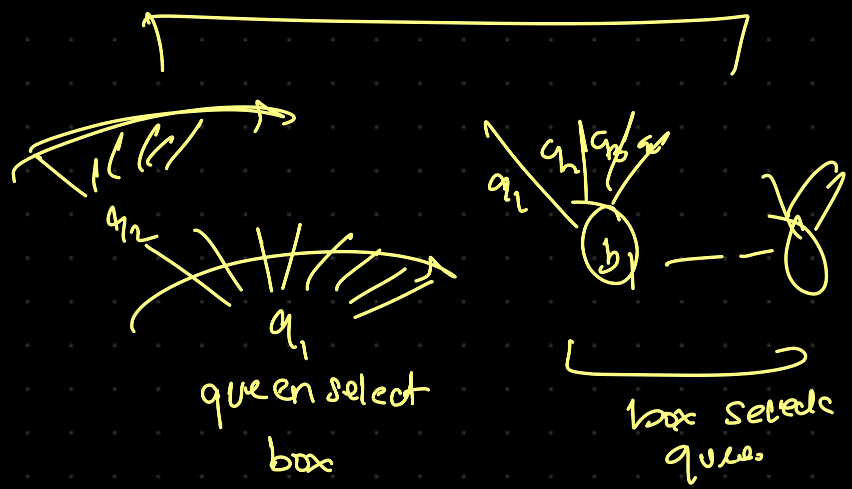
N Queens using bits:

* Recursion → for a row, we will try to place queen at every colum.
  ↳ and at next level, increment row count
* isvalid → Same Row Check ✗
  Same column check ✓
  Diagonal check ✓
  Anti diagonal check ✓



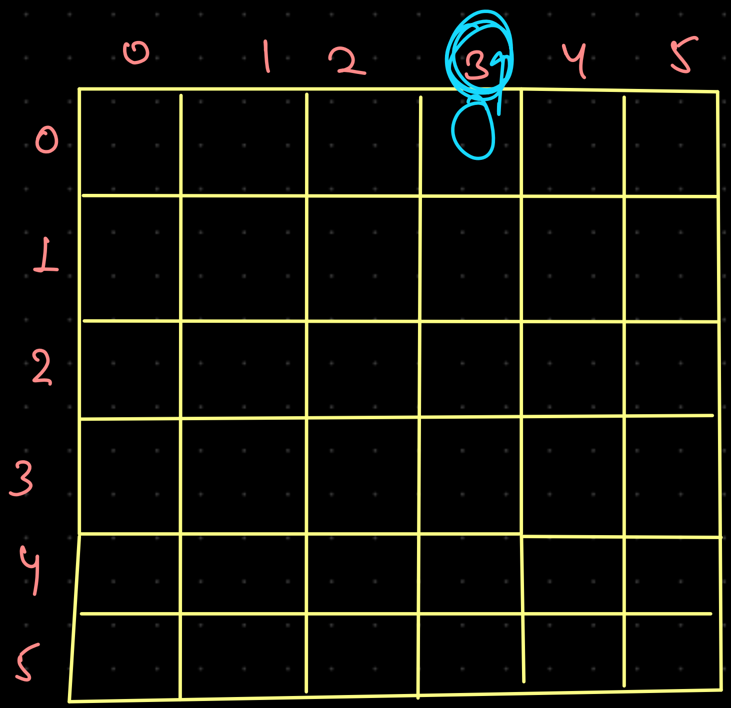
Branch and Bound is optimise version of both





validity check across

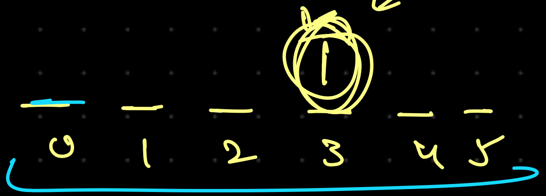
1. column check
2. Diagonal check
3. Anti diagonal check.



=

bit index → left to right

on third bit



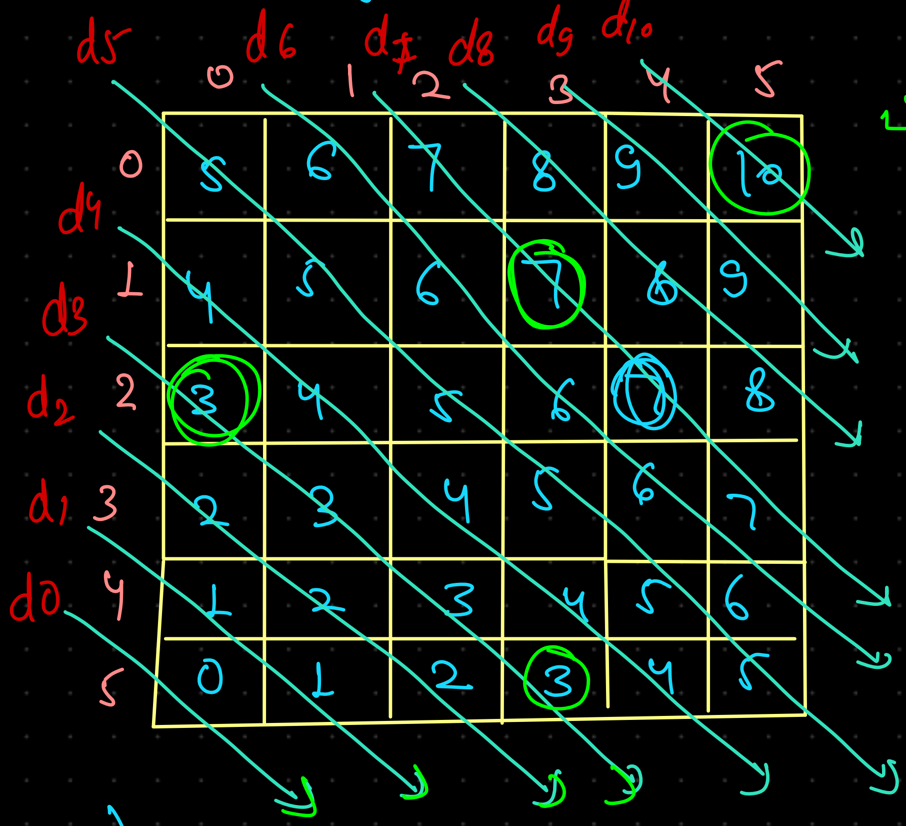
bit position of an arry

bit position

O(1) is bit is comparitively faster than O(1) of array ← theoretically equal but practically faster

5 4 3 2 1 0

Diagonal



$\underbrace{3-1}+\underbrace{6-1}$

$2+5=7$

$\underbrace{3-5}+\underbrace{6-1}$

$-2+5=3$

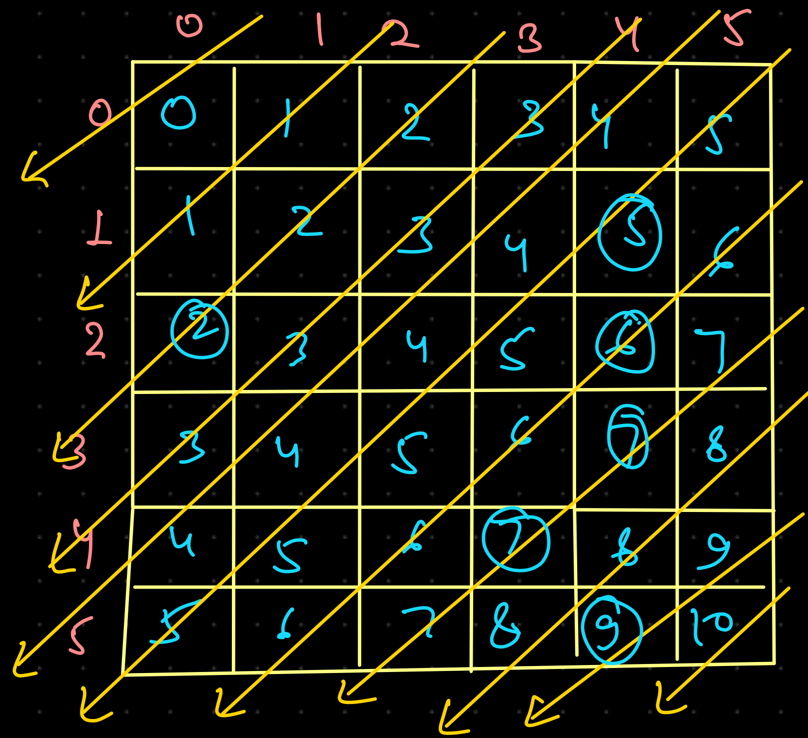
$\underbrace{5-0}+\underbrace{6-1}$

$5+5=10$

$\underbrace{0-2}+\underbrace{6-1}$

$-2+5=3$

Anti diagonal



$\frac{2+6-1}{2+5}$

diagonal No. = $\underbrace{j-i+arr.length-1}$

Anti diagonal = $\underbrace{i+j}$

Same

bit index

for safety of diagonal → 11 → size] Consider an integer & prefer this logic of bit

Total No. of diagonal = $2n-1$

$2n-1 < 64$

Is a power of 2 →

$2^0 \to 000001$

$2^1 \to 000010$

$2^2 \to 000100$

$2^3 \to 001000$

$2^4 \to 010000$

$2^5 \to 100000$

if single bit is ON

then number must be power of 2,

$n \Rightarrow 2^4 \to 010000$

$2^4 - 1 \to 001111$

$2^4 \,\&\, 2^4-1 = 000000 = 0$ 16 — 17

return [num & (num-1)] == 0;

num = 10010

num = 10001

10000

Is a power of 4 :→

① Brute Force → divide by ④ and check remainder continue, if != 0, return false.

② optimised way → bit

$$4^x \longrightarrow (2 \cdot 2)^x = (2^2)^x = 2^{2x} \longrightarrow 4^x$$

$2x$ ← Even power of 2

power of 2

	8	7	6	5	4	3	2	1	0 ← bit 8nds	
2^0 →	0	0	0	0	0	0	0	0	1	→ 4^0
2^1 →	0	0	0	0	0	0	0	1	0	
2^2 →	0	0	0	0	0	0	1	0	0	→ 4^1
2^3 →	0	0	0	0	0	1	0	0	0	
2^4 →	0	0	0	0	1	0	0	0	0	→ 4^2
2^5 →	0	0	0	1	0	0	0	0	0	
2^6 →	0	0	1	0	0	0	0	0	0	→ 4^3
2^7 →	0	1	0	0	0	0	0	0	0	
2^8 →	1	0	0	0	0	0	0	0	0	→ 4^4

2^0 →		0	0	0	0	0	0	0	1
2^2 →		0	0	0	0	0	1	0	0
2^4 →		0	0	0	1	0	0	0	0
2^6 →		0	1	0	0	0	0	0	0
2^8 →	1	0	0	0	0	0	0	0	0

In in power of 2, power is even or Even bit in ON in 2^x

the int is of type 4^x

Checks

1. check if it is valid type of 2^x → single bits ON
2. check if even bit is ON then no. is of type 4^x
   └ it is even bit

No → num = 0 0 1 0 0 0 0

	6	5	4	3	2	1	0
num =	0	0	1	0	0	0	0
bm =	1	0	1	0	1	0	1 0
num & bm =	0	0	0 0	0	0	0	0

odd bit ← all possitions are 1

16 + 4 = 20

```
if( num & bm == 0) {
    all odd bits of nums are OFF
}
```

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

first check

→ it is of type 4^x

bm = 0b 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10

Find 7n by 8 :→

$\frac{7*n}{8}$ without division and multiplication.

$2^0 + 2^3 + 2^4 + 2^5 = 57$

$x =$ 1 1 1 0 0 1 → 57

2^5 2^4 2^3 2^2 2^1 2^0

2 times

$x << 1$

1 1 1 0 0 1 0

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

$2^1 + 2^4 + 2^5 + 2^6$

$2 + 16 + 32 + 64 = 114$

3 → → 1 1 → $2^0 + 2^1 = 3$

2^1 2^0

$3 << 3$ 1 1 0 0 0 → $2^3 + 2^4$

2^4 2^3 2^2 2^1 2^0

$= 8 + 16 = 24$ ⇒ $x << y$

the x become $x * 2^y$

3×2^3

1 0 1 1 0 1 0 | 1 1 0 1 &

0

1 1 1 1

2^n

↳ How to find remainder using bits

left shift ↝ use for multiplication

$rem = ((1 << (n+1)) - 1) \,\&\, num$

$57 \rightarrow \begin{array}{cccccc} 1 & 1 & 1 & 0 & 0 & 1 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$ $\quad 2^0 + 2^3 + 2^4 + 2^5 = 57$

$\underbrace{57 >> 1}$ $\quad \begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} \rightarrow 2^2 + 2^3 + 2^4$

$\underline{num/2^1}$

$4 + 8 + 16 = 28 \leftarrow 57/2$

$\underline{\text{quotient}}$

eg $\rightarrow x = \begin{array}{ccccccccc} 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & 0 & 1 & 0 & 1 & \underbrace{1 \; 0 \; 1}_{\times} \end{array} \rightarrow \underbrace{2^0 + 2^2 + 2^3 + 2^5 + 2^7 + 2^8}$

$\hookrightarrow \underline{429}$

$x >> 3 \quad \begin{array}{cccccc} 1 & 1 & 0 & 1 & 0 & 1 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$

$\hookrightarrow 2^0 + 2^2 + 2^4 + 2^5$

$\hookrightarrow 53$

$429/2^3 \Rightarrow 429/8 \rightarrow 53$

CCD $\quad x >> y \Rightarrow x/2^y$

right shift is use for division.

$2^x - 1$ $\frac{7n}{8}$ = $\frac{8n - n}{8}$

2^x

division
multiplication ✗

Complexity – O(1)

division → using right shift.

$(\underbrace{n * 2^3}_{} - n) / 2^3$

↓

multiplication

→ using left shift

$(2^3 n - n) / 2^3$

→ $\boxed{[(n << 3) - n] >> 3}$ result