Set the kth Bit

Problem

Set kth bit in a word x to 1.

Idea

Shift and OR.

```
y = x | (1 << k);
```

Example

k = 7

x	1011110101101101
1 << k	0000000010000000
x | (1 << k)	1011110111101101

Clear the kth Bit

Problem

Clear the kth bit in a word x.

Idea

Shift, complement, and AND.

```
y = x & ~(1 << k);
```

Example

k = 7

x	1011110111101101
1 << k	0000000010000000
~(1 << k)	1111111101111111
x & ~(1 << k)	1011110101101101

Toggle the kth Bit

Problem

Flip the k th bit in a word x.

Idea

Shift and XOR.

```
y = x ^ (1 << k);
```

Example (0 → 1)

k = 7

x	1011110101101101
1 << k	0000000010000000
x ^ (1 << k)	1011110111101101

Why Learn Bit Hacks?

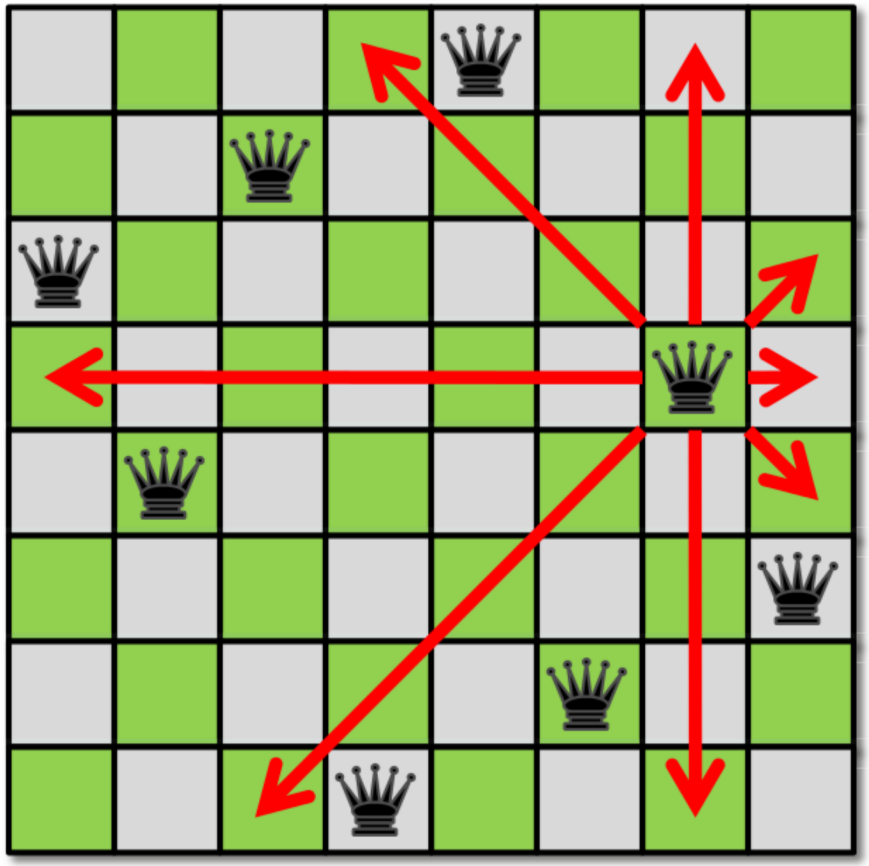
Why learn bit hacks if they don't even work?

- Because the compiler does them, and it will help to understand what the compiler is doing when you look at the assembly code.
- Because sometimes the compiler doesn't optimize, and you have to do it yourself by hand.
- Because many bit hacks for words extend naturally to bit and word hacks for vectors.
- Because these tricks arise in other domains, and so it pays to be educated about them.
- Because they're fun!

Queens Problem

Problem

Place n queens on an $n \times n$ chessboard so that no queen attacks another, i.e., no two queens in any row, column, or diagonal. Count the number of possible solutions.

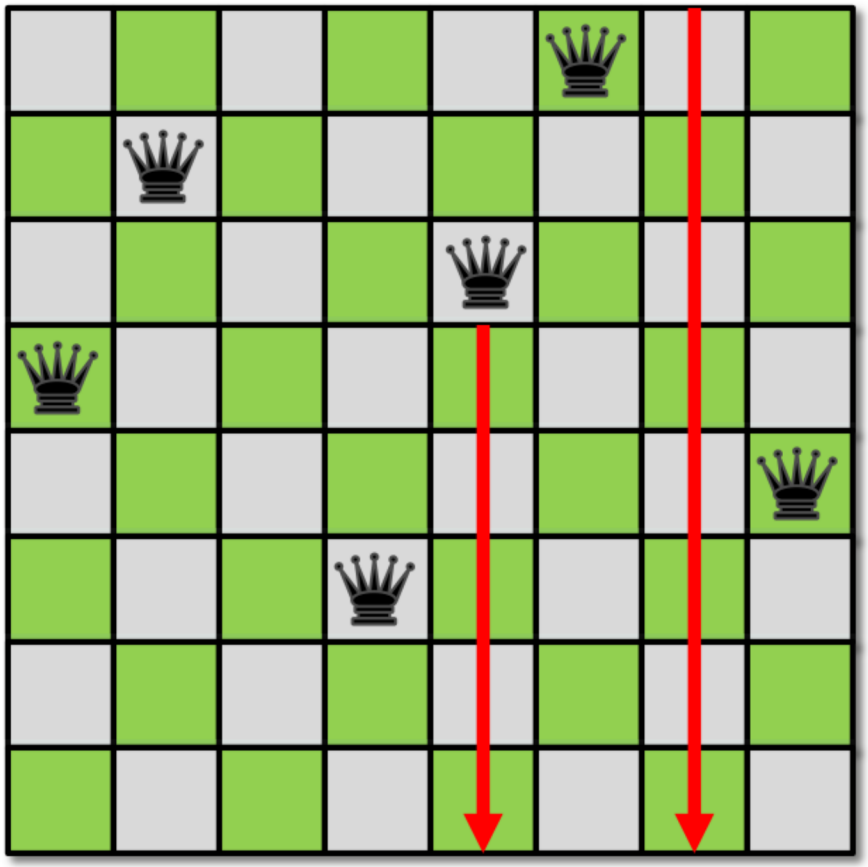


Board Representation

The backtrack search can be implemented as a simple recursive procedure, but how should the board be represented to facilitate queen placement?

- array of n^2 bytes?
- array of n^2 bits?
- array of n bytes?
- 3 bitvectors of size n, $2n-1$, and $2n-1$.

Bitvector Representation



1 1 0 1 1 1 0 1

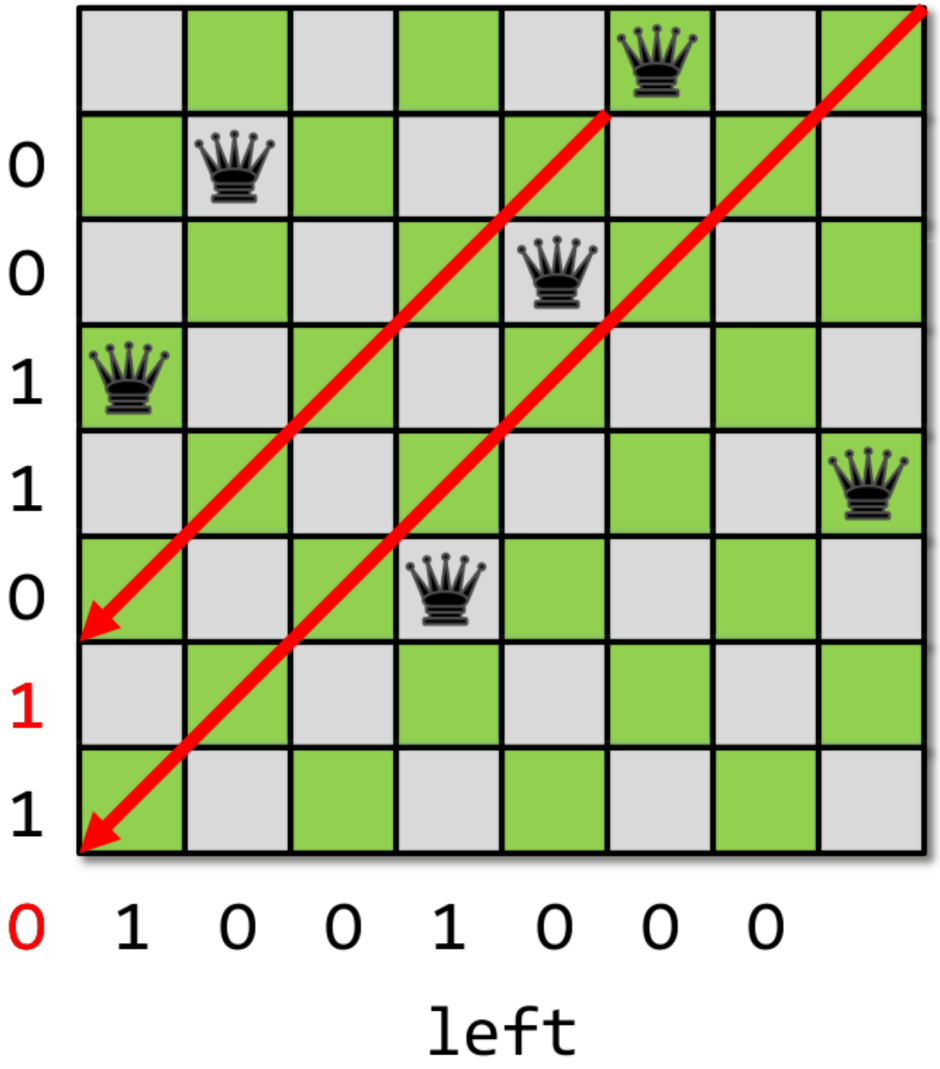
down

Placing a queen in column `c` is not safe if

```
down & (1 << c);
```

is nonzero.

Bitvector Representation

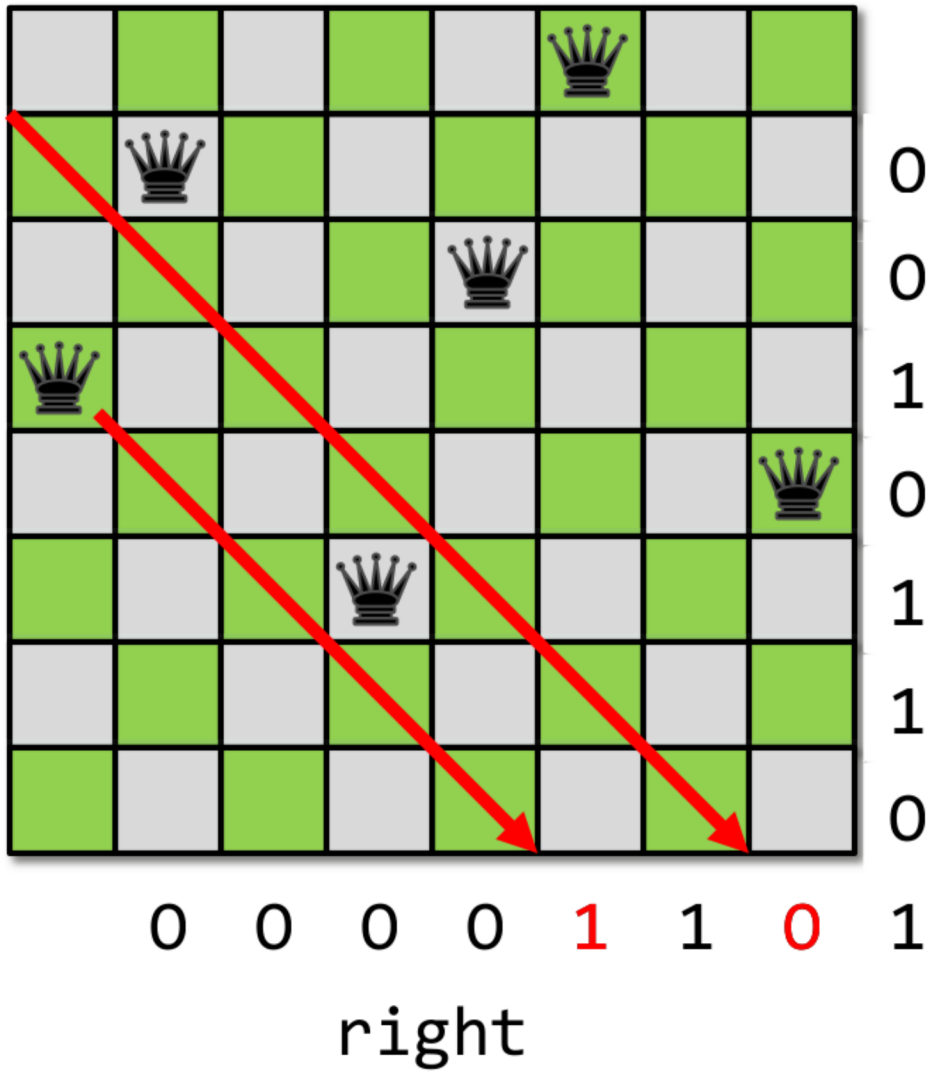


Placing a queen in row `r` and column `c` is not safe if

```
left & (1 << (r+c))
```

is nonzero.

Bitvector Representation



Placing a queen in row `r` and column `c` is not safe if

```
right & (1 << (n-1-r+c))
```

is nonzero.