# Binary Counter Increment

Put a $k$-bit number in an array $C$ of $k$ bits. LSB at $C[0]$.
Initially all 0's.

```
increment():
    i := 0
    while i < C.length and C[i] = 1:
        C[i] := 0
        i := i + 1
    if i < C.length:
        C[i] := 1
```

(For this example: modifying a bit takes $\Theta(1)$ time.)

Up to $k$ bits could be already 1. Increment takes $\Theta(k)$ time worst case. What about a sequence of $m$ increments?

# Binary Increment: Aggregate Method

- $C[0]$ is modified $m$ times
- $C[1]$ is modified $\lfloor m/2 \rfloor$ times
- $C[2]$ is modified $\lfloor m/4 \rfloor$ times
- $C[i]$ is modified $\lfloor m/2^i \rfloor$ times

Total number of modifications:

$$\begin{aligned} \sum_{i=0}^{k-1} \left\lfloor \frac{m}{2^i} \right\rfloor &< \sum_{i=0}^{\infty} \frac{m}{2^i} \\ &= 2 \cdot m \end{aligned}$$

$m$ incrememts take $O(m)$ total time. Amortized time $O(1)$.

# Binary Increment: Accounting Method

Each increment receives \$2. Prove this invariant:
\$1 savings is attached to each bit storing 1.

Initially: \$0 savings, no bit stores 1.

Increment: If each bit storing 1 has \$1 saved before:

- increment receives \$2
- change some bits from 1 to 0: spend their attached dollars (does not use the received \$2)
- may change a bit from 0 to 1: spend \$1, save \$1

Then each bit storing 1 has \$1 saved after.

Savings $\geq$ how many bits store 1's $\geq$ 0.

Amortized time $O(2)$, i.e., $O(1)$.

# Binary Increment: Potential Method

Choose $\Phi_i$ = number of bits storing 1's after $i$ increments.
Check: $\Phi_m \geq \Phi_0$ because $\Phi_0 = 0$.
Therefore can use: amortized = actual + $\Phi_i - \Phi_{i-1}$.

At each increment:

- Say, $t$ bits are changed from 1 to 0.
- In addition, may change 1 bit from 0 to 1.
- actual time $\leq t + 1$, we are changing $t$ or $t + 1$ bits
- $\Phi_i - \Phi_{i-1} \leq -t + 1$, we lost $t$ 1's and may gain back a 1

$$\begin{aligned} \text{amortized time} &= (\text{actual time}) + \Phi_i - \Phi_{i-1} \\ &\leq (t + 1) + (-t + 1) \\ &= 2 \end{aligned}$$

Amortized time $O(2)$, i.e., $O(1)$.