05.08.2024

$T(0) = T(1) \leq c$

↓ time complexity $= O(1), \Theta(1)$

$T(n) = T(n-1) + T(n-2) + \Theta(1)$

Assumption: Addition takes $b(1)$ time.

$T(n-1) \geq T(n-2)$

$\geq 2T(n-2) + \Theta(1)$ (c)

$\geq 2[2T(n-4) + \Theta(1)] + \Theta(1)$ (c, c)

$= 2^2 T(n-4) + c[2+1]$

$$\geq 2^j \cdot T(n-2j) + c\,[1+2+\cdots+j]$$

$$\geq 2^{n/2} T(n-n) + T(1) + \underbrace{c \cdot \Theta(n^2)}_{j \cong n/2}$$

$$\geq 2^{n/2} \cdot 2^{\Theta(n)}$$

$n = 20, 40,$ ~~Hopeless!~~

Can we do better ?

```
Procedure  Ifib( n )

F ← empty array of size n.

F(0) ← 0

F(1) ← 1

for j = 2 to n-1

        F(j) = F(j-1) + F(j-2).

Return  F(n-1).
```

$(\theta(1) + \theta(1))\ \theta(n)$.

fetching adding.

$= \theta(n)$ ← $10^6, 10^7$

$\geq 2^{n/2}$.

IFib.

Running time $= \theta(n)$.

running time of Fib(n) computation. $O(n)$

$\Omega(n)$.

$\theta(n)$

Remark

Assumption. Addition of two n-bit numbers has complexity $\Theta(n)$.

$2^n \geq F(n) = F(n-1) + F(n-2)$.

$\geq 2^{n/2}$.

$n/2 \leq$ # bits $\leq n$.

$\sum_{j=0}^{n} \Theta(j) = \Theta(n^2)$.

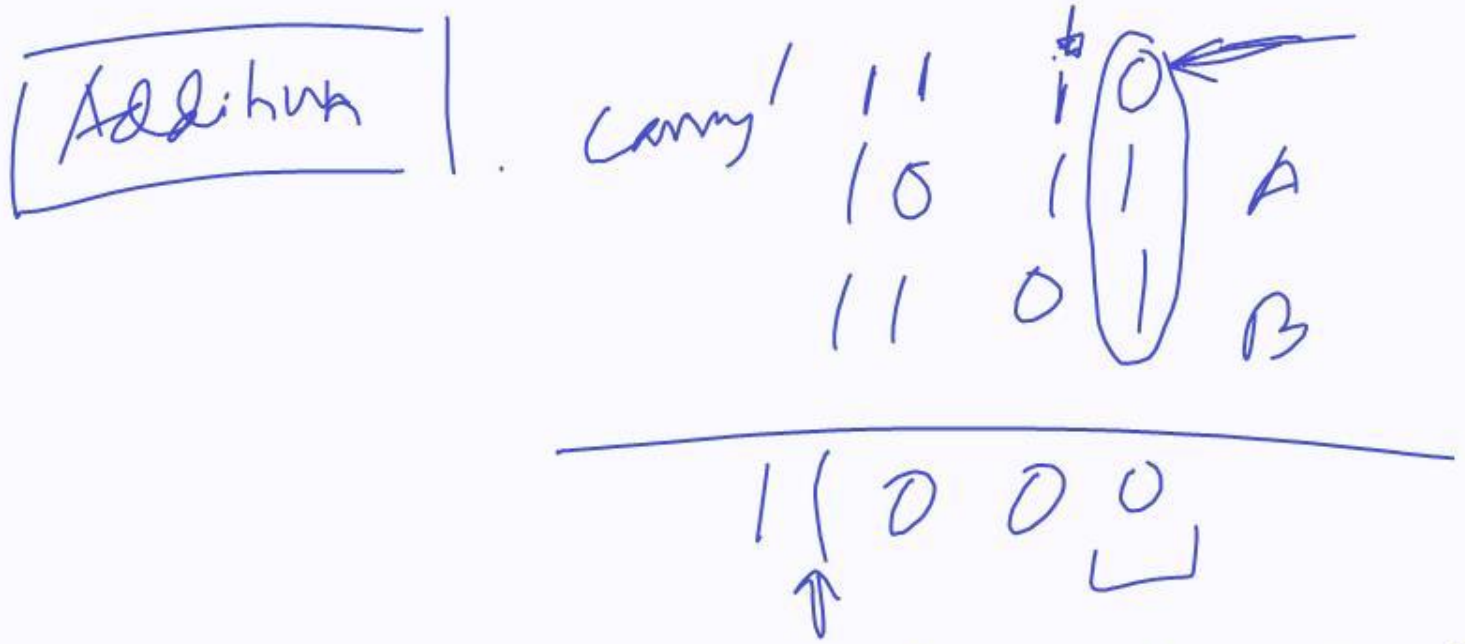
End of Module 0

- Reading Exercise: Dasgupta
- Exercise.

Module 1: Algorithm with numbers

- addition
- subtraction (later)
- multiplication
- gcd.
- modular arithmetic.

Addition



⊕ ← XOR , parity : even → 0 , odd → 1

Procedure Add (A, B)

// A, B are both n-bit numbers.

C ← 0

for i = 1 to n.

R[i] ← A[i] ⊕ B[i] ⊕ C

if (atleast 2 of A[i], B[i], C is 1)

C ← 1

else

C ← 0

time complexity

Each iteration ← $\Theta(1)$

total time $\Theta(n)$.

time complexity of algorithm → $O(n)$.

→ $\Omega(n)$.

∴ $\Theta(n)$.

Multiplication

```
      1011      A
    * 1101      B
  ------
      1011
     0000
    1011
   1011
  ------
  10001111
```

Increase of bits

If we add two n-bit numbers,

How large is the output?

— $\leq (n+1)$-bit

$$2^n - 1 + 2^n - 1 = 2^{n+1} - 2$$

$(n+1)$-bits.

1 0 1 1

1 1 0 0

---

1 . . . .

Fact: If we add two bits, we can get a 2 bit number

Fact: If we add two base b numbers, we get a 2-bit number. ($b = 2, 10, 16$)

$$(b-1)+(b-1)+(b+1) \equiv 3b-3$$

$$\equiv b+(2b-3)$$

$$(1, 2b-3)$$

Slight correction -- it should be: (3b-3)=2b+(b-3)=(2,b-3) in base-b representation for b>2.

For b=2 this is correct.

Multiplication

If we have two n-bit numbers, how large will be the result after multiplication. ?

$$(2^n - 1)(2^n - 1) \leq 2^{2n} - 2^{n+1} + 1$$

2n-bit number.

Procedure Multiply (A, B)

// A, B are two n-bit numbers.

// A[1] ---- A[n] is their array A.

// 1 indexing is used.

R ← empty of array of size 2n.

// 1 position of array is lsb.

// B: 1 0 1 1

4 3 2 1

for (j = 1 to n .

if (B[j] = 0)

A ← left shift (A).

continue

else if

R ← R + A

A ← left Shift (A).

↓

shift left & appeal a 0.

return R.

Correctness: obvious

Time complexity: $\Theta(n) \cdot \Theta(n) = \Theta(n^2)$.

Multiplication has time complexity ~~$\Theta(n^2)$~~ $O(n^2)$.

You can do better!

Subtraction: Skipped.

Division

```
Procedure Division (x, y)
// divide x/y,   x, y are +ve.
// p quotient, r ← remainder.
// x, y are n bits. q r.
if (y = 0) raise "error"
if (x = 0) return (0, 0)
end if
```

$z \leftarrow$ most $(n-1)$ significant bits of x

$(q', r') \leftarrow$ Division (z, y).

$q \leftarrow 2 \cdot q'$ ✓

$r \leftarrow 2r'$ ✓

if $(x[1] = 1)$ $r \leftarrow r + 1$ end if

if $(r > y)$

{

$q \leftarrow q + 1$

$r \leftarrow r - y$

}

end if

return (q, r)

Correctness

$$x \leftarrow 2z + b \qquad b \in \{0, 1\}$$

$$z \leftarrow q'y + r' \quad ; \quad q = 2q' + r \in \{r', r'+1\}$$

$$r \leftarrow (2r' + b) \bmod y.$$

$$q \leftarrow q' + 1$$

$$q'$$

How much time does it take. ?

$$T(n) = T(n-1) + \underbrace{\Theta(n)}_{c \cdot n}$$

$$= T(n-2) + cn + c(n-1)$$

$$\vdots$$

$$= T(n) + c[n + (n-1) + \cdots + 1]$$

$$= \Theta(1) + \Theta(n^2)$$

$$= \Theta(n^2).$$

Division has time complexity $O(n^2)$.

Modular Arithmetic

$(x + y) \bmod N$

remainder of $(x+y)$ when divided by N.

- cryptographic algorithms

Q: Compute $(X+Y)$ mod N

X, Y, N are both n-bit integers.

— $\Theta(n)$, $\Theta(n^2)$ → $\Theta(n^2)$ time!

Q: Compute $(X \cdot Y)$ mod N

$\Theta(n^2)$, $\Theta(n^2)$ → $\Theta(n^2)$ time!