

Somma Bit a Bit

Wed, 23 Mar

Supponiamo di avere due **numeri binari** memorizzati in due *array* di dimensione **n** , l'*LSB* è nella posizione n dell'*array*. Vogliamo fare la **somma bit a bit** e inserire il risultato in un *terzo* *array* di dimensione $n+1$.

Istanza d'esempio

- ▼ *Somma Bit a Bit* applicata a due *array* di esempio

carry	1	1	1	0	0	0	0	0
A[]	0	1	1	0	1	0	1	1
B[]	1	1	1	1	0	1	0	0
C[]	1	0	1	0	1	1	1	1

Valutazione tempi di esecuzione

- ▼ **Pseudocodice** dell'algoritmo

```
void Somma(A[n], B[n], C[n+1])
begin
    carry = 0                                c · 1
    for i = n down to 1
        begin
            c[i+1] = a[i]+b[i]+carry          c · n
            if c[i+1] <= 1                    c · n
                carry = 1                    c ·  $t_i$ 
            else
                c[i+1] = c[i+1]-2             c ·  $f_i$ 
                carry = 1                    c ·  $f_i$ 
        end
    end
    c[1] = carry                             c · 1
end
```

Formula per il calcolo delle **operazioni**

$$T_{sbb}(n) = 2c + 3cn + ct_i + 2cf_i$$

Caso migliore: $\nexists i \ 1 \leq i \leq n \text{ t.c. } A[i] = B[i] = 1$

$$T_m(n) = 2c + 3cn + ct_i \approx \Omega(n)$$

Caso peggiore: $A[n] = B[n] = 1 \ \forall i \ 1 \leq i \leq n \ A[i] \neq 0 \text{ or } B[i] \neq 0$

$$T_p(n) = 2c + 3cn + 2cf_i \approx O(n)$$

Tempo medio: $\Theta(n)$