Efficient Algorithms on Sparse Numbers

Jean Vuillemin

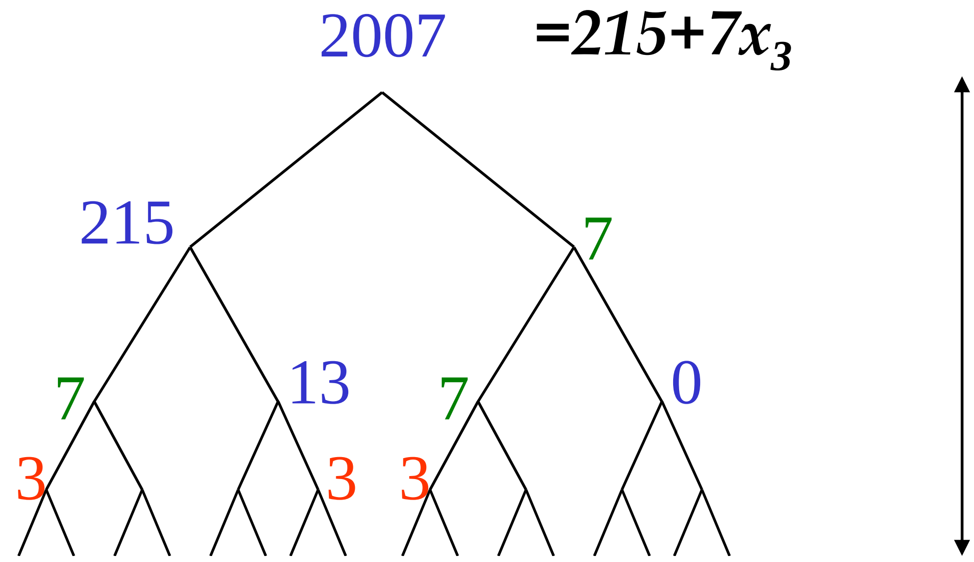
École Normale Supérieure, Paris

- **Universal Data Structure**
  - **Finite Integer**
  - **Set, Language, Polynomial …**
  - **Boolean Function – BMD (vs BDD)**
  - **Sparse & Dense Structures**
- **Store Once**
  - **Space Efficient DAG**
  - **Unit Time n=m** 2^{2^n} **…**
  - **Fast n<m -n ~n 1+n n-1** 2^n **<< >> …**
- **Compute Once**
  - **Memo 2x + - x / & | …**

2007 as a Tree

$x_3 = 2^{2^3} = 256$

$x_4 = 2^{2^4} = 65536$



$= 215 + 7x_3$

$ll(2007) = 4$

$2007 =_2 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ (0)$

$l(2007) = 11 < x_2 = 2^{2^2} = 16$

Binary length $l(n) = \lceil \log_2 n + 1 \rceil$

Binary depth $ll(n) = \lceil \log_2 \log_2 n + 1 \rceil$ 2

Integer Dichotomy

$$\mathbb{N}_0 = \{0,1\} \quad x_p = 2^{2^p}$$

$$\mathbb{N}_{p+1} = \mathbb{N}_p + x_p \mathbb{N}_p \quad \mathbb{N} = \bigcup_{p \in \mathbb{N}} \mathbb{N}_p$$

$$op(0)(and)(a,b) = ab \quad op(0)(or)(a,b) = a + b - ab$$

$$op(n+1)(f)(a,b) = op(n)(f)(a_0,b_0) + x_n op(n)(f)(a_1,b_1)$$

Code Once

$$adc(0)(a,b,c) = (s,r) \quad \{a + b + c = s + 2r\}$$

$$adc(n+1)(a,b,c) = (s_0 + x_n s_1, r)$$

$$\{(s_0,m) = adc(n)(a_0,b_0) \quad (s_1,r) = adc(n)(a_1,b_1,m)\}$$

$$maad(0)(a,b,c,d) = q_0 + x_0 q_1 \quad \{ab + c + d = q_0 + 2q_1\}$$

$$maad(n+1)(a,b,c,d) = (q_0 + x_n s_0) + x_{n+1} p \ \{$$

$$q = maad(n)(a_0,b_0,c_0,d_0)$$

$$r = maad(n)(a_0,b_1,c_1,q_1)$$

$$s = maad(n)(a_1,b_0,r_0,d_1)$$

$$p = maad(n)(a_1,b_1,s_1,r_1) \ \}$$

Integer Dichotomy

- Similar Code for XOR, OR, SUB, SHIFT, DIV, …
- Stop recursion at machine word-length

Tree vs. Array:

- Concise Recursive Code
- Proportional time & memory
- Simpler Storage Allocation

Store Once

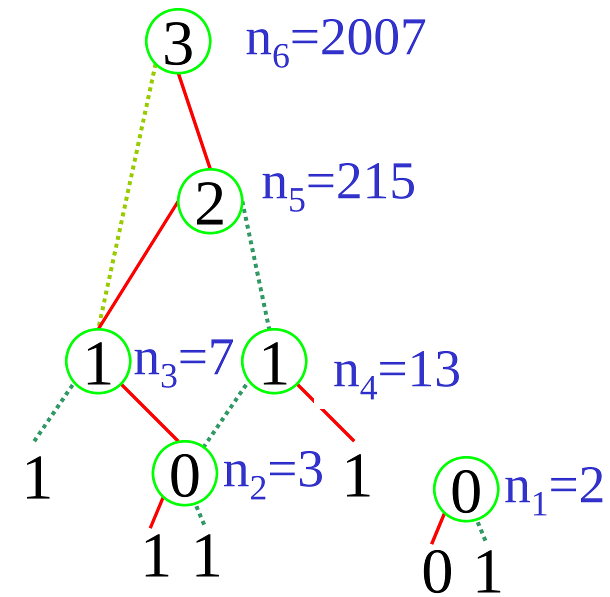
$2007 =_2 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ (0)$

$n_6 = n_5 + x_3 n_3 = 2007$

$n_5 = n_3 + x_2 n_4 = 215$

$n_3 = n_2 + x_1 = 7 \qquad n_4 = 1 + x_1 n_1 = 13$

$n_1 = 0 + x_0 = 2 \qquad n_2 = 1 + x_0 = 3$



6 lines of SSA code ⟺ ***6 nodes in DAG***

Compare

$$sign(n-m)=\begin{cases}1 \Leftrightarrow n>m\\ 0 \Leftrightarrow n=m\\ -1 \Leftrightarrow n<m\end{cases}$$

Array at worst *l(n)* operations.

DAG at worst *ll(n)* operations:

$$sign(n-m)=\begin{cases}n=m & \Rightarrow & 0\\ n_p \neq m_p & \Rightarrow & sign(n_p-m_p)\\ n_1 \neq m_1 & \Rightarrow & sign(n_1-m_1)\\ & \Rightarrow & sign(n_0-m_0)\end{cases}$$

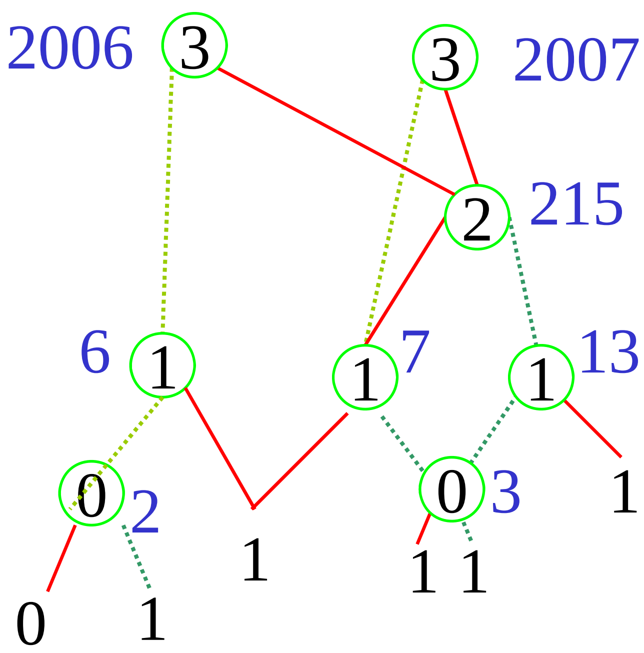
Decrement

n-1 in *ll(n)* operations:

$$n = t(g, p, d)$$
$$m = n - 1$$

$$g = 0 \quad \Rightarrow \quad m = t(\mu p, p, d-1)$$
$$g \neq 0 \quad \Rightarrow \quad m = t(g-1, p, d)$$

Table: $\mu p = 2^{2^p} - 1$



Implies *s(n,n-1)<s(n)+ll(n)*

ALU $op \in \{+, -, \oplus, \cap, \ldots\}$ operation *n* op *k* with sparse constant $k \in \{1, 2, 4, \ldots\}$.

Huge

$$b_0 = 1$$

$$b_{n+1} = t(b_n, b_n, b_n)$$

$$b_n > 2 \uparrow 2n$$

$$l(b_{n+1}) = 2^{b_n}$$

$$ll(b_{n+1}) = b_n$$

$$v(b_n) = 2^n$$

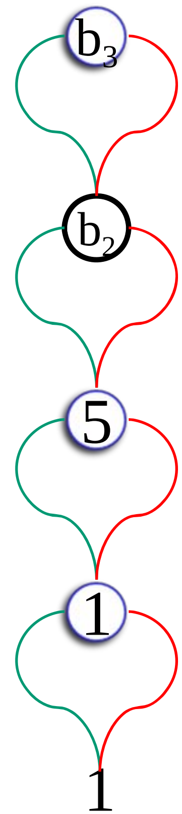
$$s(b_n) = n$$

$$s(1 + b_n) = 2n$$

$$s(b_n - 1) = 2n - 1$$

$$s(2^{b_n}) = 2^n$$

$$s(l(b_{n+1})) = 2^n$$



$$b_4 = b_3 (1+2^{2^{b_3}})$$

$$b_3 = b_2 (1+2^{2^{b_2}}) > 2^{2^{2^{2^{2^{2^{2^{2}}}}}}}$$

$$b_2 = 5 (1+2^{32}) = 21474836485$$

$$b_1 = 5$$

Demo

Compute Once

$$mul_2(0) = 0 \quad mul_2(1) = \mathrm{t}(0,0,1)$$

$$mul_2\ t(l,p,h) = \mathrm{t}(mul_2(l), p, mul_2(h))$$

Naive doubling takes *l(n)* operations.

Time to traverse DAG can be exponential in size

Implement ***mul*$_2$** as memo function:

Computed values temporarily stored & retrieved when possible;

Compute all ***exactly once!***

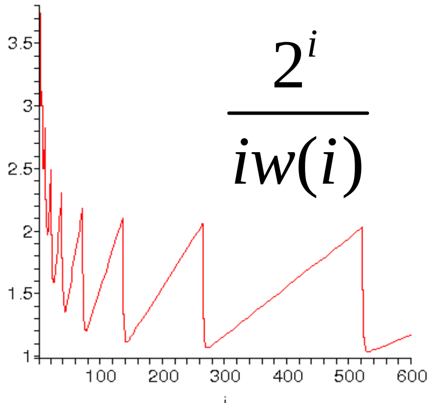
mul space (& time)**:** $s(n \times m) < s(n) \times s(m)$ ***No*** Karatsuba!

Over 10^6 DAGs operationally near x_{1024} are ***very sparse***:

s(n)<ll(n).

Worst, Average Case

$$s(n) \le v(n) \le l(n)$$



$$\frac{2^i}{iw(i)}$$

$$s(n) < \frac{2l(n)}{ll(n) - lll(n)}$$

n sparse ⇔ s(***n***)=O(ll(***n***))⇔low entropy.

$$s(1 \cdots n) = n \qquad a(1 \cdots n) = nl(n)$$

2007 as a Boolean Expression

$x_0 = 2 \quad x_1 = 4 \quad x_2 = 16 \quad x_3 = 256 \quad x_p = 2^{2^p}$

F(2007) $= {}_2 1\,1\,1\,0\,1\,0\,1\,1\,1\,1\,1$

$= 1\,1\,1\,0\,1\,0\,1\,1 + x_3\,1\,1\,1$

$= (((1 + x_0) + x_1) + x_2(1 + x_1(1 + x_0))) + x_3((1 + x_0) + x_1)$

BMD package when restricted to Boolean operations

Duality between BMD & BDD

2007 as a Set

$$2007 = \sum_{s \in S} 2^s$$

S(2007)= {0 1 2 4 6 7 8 9 10}

2007 as a Language

***L(2007)**= {0000 1000 0100 0010 0110 1110 0001 1001 0101}*

Versatile Data Structure

Number = ll(n) 2^{2^n} t(l,p,h) < -n ~n s+n l(n) 2^n ≪ ≫ + - × ÷ mod

Set = ∈ min max insert delete sort median ∪ ∩ ⊂

Series z z^- ↑ ↓ ⊕ ⊗

Language 0 1 ¬ + • e^n 0^- 1^-

Conclusion

Sparse Numbers (entropy near 0)

DAG size << binary length.

DAG time at worst quadratic in size.

Dense Numbers (entropy near 1)

DAG size < binary length.

DAG time < c bit-array time.

To Do (call for cooperation):

Achieve c<2.

Tie in efficient storage de/allocation.

Sparse matrices, images, video, files, …