Some Great Theoretical Ideas in Computer Science

15-251

Ancient Wisdom: On Raising A Number To A Power
Lecture 14 (February 28, 2008)

Egyptian Multiplication

The Egyptians used decimal numbers but multiplied and divided in binary

a x b By Repeated Doubling

b has n-bit representation: $b_{n-1}b_{n-2}...b_1b_0$

Starting with a, repeatedly double largest number so far to obtain: $a, 2a, 4a, ..., 2^{n-1}a$

Sum together all the $2^k a$ where $b_k = 1$

$b = b_02^0 + b_12^1 + b_22^2 + ... + b_{n-1}2^{n-1}$

$ab = b_02^0a + b_12^1a + b_22^2a + ... + b_{n-1}2^{n-1}a$

$2^k a$ is in the sum if and only if $b_k = 1$
Wait!
How did the Egyptians do the part where they converted \( b \) to binary?

They used repeated halving to do base conversion!

**Egyptian Base Conversion**

Output stream will print right to left

```plaintext
Input X;
repeat {
  if (X is even) then print 0;
  else
    {X := X-1; print 1;}
  X := X/2;
} until X=0;
```

Sometimes the Egyptians combined the base conversion by halving and multiplication by doubling into a single algorithm.
### $70 \times 13$

**Rhind Papyrus [1650 BC]**

<table>
<thead>
<tr>
<th>Doubling</th>
<th>Halving</th>
<th>Odd?</th>
<th>Running Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>13</td>
<td>*</td>
<td>70</td>
</tr>
<tr>
<td>140</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>280</td>
<td>3</td>
<td>*</td>
<td>350</td>
</tr>
<tr>
<td>560</td>
<td>1</td>
<td>*</td>
<td>910</td>
</tr>
</tbody>
</table>

Binary for 13 is $1101 = 2^3 + 2^2 + 2^0$

$70 \times 13 = 70 \times 2^3 + 70 \times 2^2 + 70 \times 2^0$

---

### $30 \times 5$

<table>
<thead>
<tr>
<th>Doubling</th>
<th>Halving</th>
<th>Odd?</th>
<th>Running Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>30</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>*</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>*</td>
<td>70</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

---

### $184 / 17$

**Rhind Papyrus [1650 BC]**

<table>
<thead>
<tr>
<th>Doubling</th>
<th>Powers of 2</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>2</td>
<td>*</td>
</tr>
<tr>
<td>68</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>8</td>
<td>*</td>
</tr>
</tbody>
</table>

$184 = 17 \times 8 + 17 \times 2 + 14$

$184/17 = 10$ with remainder $14$

---

This method is called
“Egyptian Multiplication / Division”
or
“Russian Peasant Multiplication / Division”
Standard Binary Multiplication = Egyptian Multiplication

```
*********
× 1101
*********
*********
*********
*********
```

Important Technique:

Abstraction
Abstract away the inessential features of a problem or solution

```
= ♫ ♫ ♫
```

Powering By Repeated Multiplication

**Input:** a, n
**Output:** Sequence starting with a, ending with aⁿ, such that each entry other than the first is the product of two previous entries

```
b:=a^8
b:=a*a
b:=b*b
b:=b*b
b:=b*b
b:=b*b
b:=b*b
b:=b*b
b:=b*a
```

This method costs only 3 multiplications. The savings are significant if b:=a^8 is executed often.
Example

Input:  a,5
Output:  a, a^2, a^3, a^4, a^5
or
Output:  a, a^2, a^3, a^5
or
Output:  a, a^2, a^4, a^5

Given a constant n, how do we implement b:=a^n with the fewest number of multiplications?

Definition of M(n)

M(n) = Minimum number of multiplications required to produce a^n from a by repeated multiplication

What is M(n)? Can we calculate it exactly? Can we approximate it?
Very Small Examples

What is $M(1)$?
$M(1) = 0 \quad [a]$ 

What is $M(0)$?
Not clear how to define $M(0)$

What is $M(2)$?
$M(2) = 1 \quad [a, a^2]$

$M(8) = ?$
$a, a^2, a^4, a^8$ is one way to make $a^8$ in 3 multiplications

What does this tell us about the value of $M(8)$?

$M(8) \leq 3$

3 ≤ $M(8)$ ≤ 3

3 ≤ $M(8)$ ≤ 3 by exhaustive search

There are only two sequences with 2 multiplications. Neither of them make 8:
a, a^2, a^3 and a, a^2, a^4

$M(8) = 3$
**Abstraction**

Abstract away the inessential features of a problem or solution.

**Representation:**
Understand the relationship between different representations of the same idea.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the more essential representation of $M(n)$?

The “$a$” is a red herring

$a^xa^y$ is $a^{x+y}$

Everything besides the exponent is inessential. This should be viewed as a problem of repeated addition, rather than repeated multiplication.

**Addition Chains**

$M(n) =$ Number of stages required to make $n$, where we start at 1 and in each stage we add two previously constructed numbers.

**Examples**

Addition Chain for 8:

1 2 3 5 8

Minimal Addition Chain for 8:

1 2 4 8
Addition Chains Are a Simpler Way To Represent The Original Problem

| Abstraction: Abstract away the inessential features of the problem or solution |
| Representation: Understand the relationship between different representations of the same idea |
| 1 | 2 | 3 |

Addition Chains For 30

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>24</th>
<th>28</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

M(30) = ?

? ≤ M(30) ≤ 6

? ≤ M(n) ≤ ?
Binary Representation

Let $B_n$ be the number of 1s in the binary representation of $n$

E.g.: $B_5 = 2$ since $5 = (101)_2$

Proposition: $B_n \leq \lfloor \log_2 (n) \rfloor + 1$

(It is at most the number of bits in the binary representation of $n$)

Binary Method

(Repeated Doubling Method)

Phase I (Repeated Doubling)

For $\lfloor \log_2 (n) \rfloor$ stages:
Add largest so far to itself

(1, 2, 4, 8, 16, ...)

Phase II (Make $n$ from bits and pieces)

Expand $n$ in binary to see how $n$ is the sum of $B_n$ powers of 2. Use $B_n - 1$ stages to make $n$ from the powers of 2 created in phase I

Total cost: $\lfloor \log_2 n \rfloor + B_n - 1$

---

Binary Method Applied To 30

Phase I

1, 2, 4, 8, 16  Cost: 4 additions

Phase II

30 = (11110)\text{$_2$}

2 + 4 = 6

6 + 8 = 14

14 + 16 = 30  Cost: 3 additions

$M(n) \leq \lfloor \log_2 n \rfloor + B_n - 1 \leq 2 \lfloor \log_2 n \rfloor$
Rhind Papyrus [1650 BC]

What is 30 x 5?

Repeated doubling is the same as the Egyptian binary multiplication.

Addition chain for 30:
- 1
- 2
- 4
- 8
- 16
- 24
- 28
- 30

Start at 5 and perform same additions as chain for 30:
- 5
- 10
- 20
- 40
- 80
- 120
- 140
- 150

The Egyptian Connection

A shortest addition chain for n gives a shortest method for the Egyptian approach to multiplying by the number n.

The fastest scribes would seek to know M(n) for commonly arising values of n.

Rhind Papyrus [1650 BC]

Actually used faster chain for 30 x 5:

- 1
- 2
- 4
- 8
- 10
- 20
- 40
- 50
- 100
- 150

? \leq M(30) \leq 6

? \leq M(n) \leq 2 \left\lfloor \log_2 n \right\rfloor
A Lower Bound Idea

You can’t make any number bigger than $2^n$ in $n$ steps

1 2 4 8 16 32 64 . . .

or is this a failure of imagination?

Let $S_k$ be the statement that no $k$ stage addition chain contains a number greater than $2^k$

Base case: $k=0$. $S_0$ is true since no chain can exceed $2^0$ after 0 stages

$\forall k > 0, \ S_k \Rightarrow S_{k+1}$

At stage $k+1$ we add two numbers from the previous stage

From $S_k$ we know that they both are bounded by $2^k$

Hence, their sum is bounded by $2^{k+1}$: No number greater than $2^{k+1}$ can be present by stage $k+1$

Change Of Variable

All numbers obtainable in $m$ stages are bounded by $2^m$. Let $m = \log_2(n)$

Thus, all numbers obtainable in $\log_2(n)$ stages are bounded by $n$

$$M(n) \geq \lceil \log_2 n \rceil$$

$\text{?} \leq M(30) \leq 6$

$$\lceil \log_2 n \rceil \leq M(n) \leq 2 \lceil \log_2 n \rceil$$
Theorem: $2^i$ is the largest number that can be made in $i$ stages, and can only be made by repeated doubling

Proof by Induction:

Base $i = 0$ is clear

To make anything as big as $2^i$ requires having some $X$ as big as $2^{i-1}$ in $i-1$ stages

By I.H., we must have all the powers of 2 up to $2^{i-1}$ at stage $i-1$. Hence, we can only double $2^{i-1}$ at stage $i$.

---

$5 < M(30)$

Suppose that $M(30)=5$

At the last stage, we added two numbers $x_1$ and $x_2$ to get 30

Without loss of generality (WLOG), we assume that $x_1 \geq x_2$

Thus, $x_1 \geq 15$

By doubling bound, $x_1 \leq 16$

But $x_1 \neq 16$ since there is only one way to make 16 in 4 stages and it does not make 14 along the way. Thus, $x_1 = 15$ and $M(15)=4$

---

Suppose $M(15) = 4$

At stage 3, a number bigger than 7.5, but not more than 8 must have existed

There is only one sequence that gets 8 in 3 additions: 1 2 4 8

That sequence does not make 7 along the way and hence there is nothing to add to 8 to make 15 at the next stage

Thus, $M(15) > 4$  CONTRADICTION

---

$M(30)=6$
Factoring Bound

\[ M(a \times b) \leq M(a) + M(b) \]

Proof:

Construct \( a \) in \( M(a) \) additions

Using \( a \) as a unit follow a construction method for \( b \) using \( M(b) \) additions.

In other words, each time the construction of \( b \) refers to a number \( y \), use the number \( ay \) instead.

Example

\[
\begin{align*}
M(30) &= 6 \\
\left\lceil \log_2 n \right\rceil &\leq M(n) \leq 2 \left\lfloor \log_2 n \right\rfloor \\
45 &= 5 \times 9 \\
M(5) &= 3 \quad [1 \ 2 \ 4 \ 5] \\
M(9) &= 4 \quad [1 \ 2 \ 4 \ 8 \ 9] \\
M(45) &\leq 3 + 4 \quad [1 \ 2 \ 4 \ 5 \ 10 \ 20 \ 40 \ 45]
\end{align*}
\]

Corollary (Using Induction)

\[ M(a_1 a_2 a_3 \ldots a_n) \leq M(a_1) + M(a_2) + \ldots + M(a_n) \]

Proof:

For \( n = 1 \) the bound clearly holds

Assume it has been shown for up to \( n-1 \)

Now apply previous theorem using

\( A = a_1 a_2 a_3 \ldots a_{n-1} \) and \( b = a_n \) to obtain:

\[ M(a_1 a_2 a_3 \ldots a_n) \leq M(a_1 a_2 a_3 \ldots a_{n-1}) + M(a_n) \]

By inductive assumption,

\[ M(a_1 a_2 a_3 \ldots a_{n-1}) \leq M(a_1) + M(a_2) + \ldots + M(a_{n-1}) \]
More Corollaries

Corollary: $M(a^k) \leq kM(a)$

Corollary: $M(p_1^{n_1} p_2^{n_2} \ldots p_r^{n_r}) \leq \alpha_1 M(p_1) + \alpha_2 M(p_2) + \ldots + \alpha_r M(p_r)$

Does equality hold?

$M(33) < M(3) + M(11)$

$M(3) = 2\quad [1\ 2\ 3]$  
$M(11) = 5\quad [1\ 2\ 3\ 5\ 10\ 11]$  
$M(3) + M(11) = 7$  
$M(33) = 6\quad [1\ 2\ 4\ 8\ 16\ 32\ 33]$  

The conjecture of equality fails!

Conjecture: $M(2n) = M(n) + 1$  
(A. Goulard)

A fastest way to an even number is to make half that number and then double it

Proof given in 1895 by E. de Jonquieres in L’Intermédiaire des Mathématiques, volume 2, pages 125-126

FALSE! $M(191) = M(382) = 11$

Furthermore, there are infinitely many such examples

Open Problem

Is there an $n$ such that:  
$M(2n) < M(n)$
Conjecture

Each stage might as well consist of adding the largest number so far to one of the other numbers

First Counter-example: 12,509
[1 2 4 8 16 17 32 64 128 256 512 1024 1041 2082 4164 8328 8345 12509]

Open Problem

Prove or disprove the Scholz-Brauer Conjecture:

\[ M(2^n - 1) \leq n - 1 + B_n \]

(The bound that follows from this lecture is too weak: \[ M(2^n - 1) \leq 2n - 1 \])

High Level Point

Don’t underestimate “simple” problems. Some “simple” mysteries have endured for thousand of years

Egyptian Multiplication
Raising To A Power
Minimal Addition Chain
Lower and Upper Bounds
Repeated doubling method

Here’s What You Need to Know…