FUN AT THE OFFICE #921

How Computers Work

Search ID: jsin303
Today’s true story… The E.T. Ride at Universal Studios
Please start HW early

• Friday office hours = :^)
“Hard” Problems

Snowplows of Northern Minnesota (aka the “Vertex Cover Problem”)

Brute-force? Greed? DP?
## $n^2$ and $n^3$ versus $2^n$

The Ran-O-Matic performs $10^9$ operations/sec

<table>
<thead>
<tr>
<th></th>
<th>n = 10</th>
<th>n = 30</th>
<th>n = 50</th>
<th>n = 70</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n^2$</td>
<td>100</td>
<td>900</td>
<td>2500</td>
<td>4900</td>
</tr>
<tr>
<td></td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
</tr>
<tr>
<td>$n^3$</td>
<td>1000</td>
<td>27000</td>
<td>125K</td>
<td>343K</td>
</tr>
<tr>
<td></td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
</tr>
<tr>
<td>$2^n$</td>
<td>1024</td>
<td>$10^9$</td>
<td>13 days</td>
<td>37 thousand years</td>
</tr>
<tr>
<td></td>
<td>&lt; 1 sec</td>
<td>1 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assuming computers double in speed every year, let’s just wait 10 years!
The Traveling Salesperson Problem

San Francisco  
Claremont  
New York  
Moscow  
Paris

242  
442  
2142  
1342  
2642  
1942  
742

Brute Force? Greed? DP?
The Hamiltonian Cycle Problem
Here as he walked by
on the 16th of October 1843
Sir William Rowan Hamilton
in a flash of genius discovered
the fundamental formula for
quaternion multiplication
\[ i^2 = j^2 = k^2 = ijk = -1 \]
& cut it on a stone of this bridge
Snowplows and Travelling Salesperson Revisited!

Tens of thousands of other known problems go in this cloud!!

- Snowplow Problem
- Travelling Salesperson Problem
- Protein Folding
- Sodoku

NP-hard problems
I can’t find an efficient algorithm, I guess I’m just too dumb.

I can’t find an efficient algorithm, because no such algorithm is possible!

I can’t find an efficient algorithm, but neither can all these famous people!

Millennium Problems

Yang–Mills and Mass Gap
Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

Riemann Hypothesis
The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation
This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.
“Recent” NP Excitement!

Vinay Deolalikar, HP Labs

August 6, 2010

1. Introduction

The $P \neq NP$ question is generally considered one of the most important and far reaching questions in contemporary mathematics and computer science. The origin of the question seems to date back to a letter from Gödel to Von Neumann in 1956 [Sip92]. Formal definitions of the class $NP$ awaited work by Edmonds [Edm65], Cook
A Solution of the P versus NP Problem

Norbert Blum

(Submitted on 11 Aug 2017 (v1), last revised 30 Aug 2017 (this version, v2))

Berg and Ulfberg and Amano and Maruoka have used CNF–DNF–approximators to prove exponential lower bounds for the monotone network complexity of the clique function and of Andreev's function. We show that these approximators can be used to prove the same lower bound for their non–monotone network complexity. This implies P not equal NP.
Science

P≠NP proof fails, Bonn boffin admits

Norbert Blum says his proposed solution doesn't work

By Thomas Claburn in San Francisco 31 Aug 2017 at 19:16  39  SHARE ▼

Computer science boffin Norbert Blum has acknowledged that his P≠NP proof is incorrect, as a number of experts anticipated.
Where we’ve been and where we’re going...

• Functional Programming
  – Short and powerful programs using recursion and higher-order functions

• How computers work
  – Representing data, building a small computer out of circuits, programming that computer in its native language

• Writing bigger programs
  – Object-oriented design, AI concepts, games!

• From hard to impossible problems!

• End-of-course projects
It all starts with representing data in binary!

What is the number 4312?

\[
\begin{array}{cccc}
10^3 & 10^2 & 10^1 & 10^0 \\
4 & 3 & 1 & 2
\end{array}
\]

What is this number in base 20?

\[
\begin{array}{cccc}
20^2 & 20^1 & 20^0 \\
1 & 3 & 2
\end{array}
\]

Now we’re using powers of 20.

What is this number in base 2?

\[
\begin{array}{cccc}
2^2 & 2^1 & 2^0 \\
1 & 1 & 0
\end{array}
\]

I suspect that it’s the number after 4311…

Olmec number representation in base 20 (East Mexico 1200 BC-600 AD)

Olmec relief from http://www.meta-religion.com
Arbitrary Bases (base “b”)

When using base $b$, the digits permitted are:

What is 5 in…
- base 2?
- base 3?
- base 4?
- base 5?
- base 6?
- base 42?

We write:

$$101_2 = 12_3 = 11_4 = 10_5 = 5_6 = 5_{10} = 5_{42}$$

The subscript indicates the base.
Is There Such a Thing as Base 1?

Unary!

$1^3 \ 1^2 \ 1^1 \ 1^0$

Now we’re using powers of 1 (Weird!)

Technically, we should use 0 as our digit!
Comparing Representations in Different Bases

Consider the number $10^9$ in base 1, 2, 3, 10, and 20:

**Base 1:** 11111111111111111111111111111111111111111111...

**Base 2:** 111011100110101100101000000000

**Base 3:** 212020020002101001

**Base 10:** 1000000000

**Base 20:** FCA0000

At ten “1’s” per inch, this will be...

Does FCA stand for “Friendly Cuddly Alien?”
Comparing Representations in Different Bases

Consider the number $10^9$ in base 1, 2, 3, 10, and 20:

Base 1: \[11111111111111111111111111111111111111111111\ldots\]

At ten “1’s” per inch, this will be \textbf{1578 miles long}!

Base 2: \[11101110011010110010100000000\]

Base 3: \[212020020002101001\]

Base 10: \[1000000000\]

Base 20: \textbf{FCA0000}

Does FCA stand for “Friendly Cuddly Alien?”
Converting Between Bases

Convert $1101_2$ to base 10

Convert $25_{10}$ to base 2

The digits 0 and 1 are referred to as “bits” - that’s short for “binary digits”
The “Power” of Shifting!

“Left Shifting”

\[
\begin{array}{cccc}
10^3 & 10^2 & 10^1 & 10^0 \\
\hline
4 & 2 & \quad & 0 \\
4 & 2 & \quad & 0 \\
\end{array}
\]

\[10^3 10^2 10^1 10^0\]

\[\text{Left Shifting}\]

\[2^3 2^2 2^1 2^0\]

\[\begin{array}{cccc}
2^3 & 2^2 & 2^1 & 2^0 \\
\hline
1 & 1 & \quad & 0 \\
1 & 1 & \quad & 0 \\
\end{array}\]

\[2^3 2^2 2^1 2^0\]

\[= 3_{10}\]

\[= ?\]

“Right Shifting”

\[
\begin{array}{cccc}
10^3 & 10^2 & 10^1 & 10^0 \\
\hline
4 & 5 & 7 & \quad \\
4 & 5 & 7 & \quad \\
\end{array}
\]

\[10^3 10^2 10^1 10^0\]

\[\text{Right Shifting}\]

\[2^3 2^2 2^1 2^0\]

\[\begin{array}{cccc}
2^3 & 2^2 & 2^1 & 2^0 \\
\hline
1 & 0 & 1 & \quad \\
1 & 0 & 1 & \quad \\
\end{array}\]

\[2^3 2^2 2^1 2^0\]

\[= 5_{10}\]

\[= ?\]
Base Conversion, Part Deux

25_{10} = ?_{2}
Addition

Base 10 Addition

\[
\begin{array}{ccc}
10^2 & 10^1 & 10^0 \\
4 & 3 & \\
+ & 8 & 9 \\
\end{array}
\]
Addition

Base 10 Addition

\[
\begin{array}{c}
10^2 & 10^1 & 10^0 \\
\hline
4 & 3 & \\
+ & 8 & 9 \\
\hline
1 & 2
\end{array}
\]

That’s a “10”
Addition

Base 10 Addition

\[
\begin{array}{c|c|c|c|c|c|c}
10^2 & 10^1 & 10^0 \\
\hline
1 & 4 & 3 \\
+ & 8 & 9 \\
\hline
& & 2 \\
\end{array}
\]

Move the “1” to the tens place.
Addition

Base 10 Addition

\[
\begin{array}{ccc}
10^2 & 10^1 & 10^0 \\
\hline
1 & 4 & 3 \\
+ & 8 & 9 \\
\hline
13 & 2 \\
\end{array}
\]

Done!
Addition

Base 10 Addition

\[
\begin{array}{c c c}
10^2 & 10^1 & 10^0 \\
1 & 4 & 3 \\
+ & 8 & 9 \\
\hline
13 & 2 \\
\end{array}
\]

Try it in base 2!

Base 2 Addition

\[
\begin{array}{c c c}
2^2 & 2^1 & 2^0 \\
1 & 0 & 1 \\
+ & 0 & 1 \\
\hline
1 & 1 & 1 \\
\end{array}
\]
Multiplication

Base 10 Multiplication

\[
\begin{array}{c}
10^2 & 10^1 & 10^0 \\
3 & 4 & 1 \\
\times & 1 & 0 & 2 \\
\hline
6 & 8 & 2 \\
0 & 0 & 0 \\
+ & 3 & 4 & 1 \\
\hline
3 & 4 & 7 & 8 & 2
\end{array}
\]

Base 2 Multiplication

\[
\begin{array}{c}
2^2 & 2^1 & 2^0 \\
1 & 1 & 1 \\
\times & 1 & 0 & 1 \\
\hline
1 & 0 & 1
\end{array}
\]
Aside: Multiplication with Russian Peasants

Compute \(21 \times 6:\)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>1</td>
<td>96</td>
</tr>
</tbody>
</table>

(Translation: “Hello American Students!”)

(Thanks to Danny Gorelik for the Russian translations!)
Aside: Multiplication with Russian Peasants

Compute $21 \times 6$:

\[
\begin{array}{ccc}
21 & 6 \\
10 & 12 \\
5 & 24 \\
2 & 48 \\
1 & 96 \\
\end{array}
\]

$6 + 24 + 96 = 126$

(Translation: “Why does this work?”)
Aside: Multiplication with Russian Peasants

**Compute** $21 \times 6$:

\[
\begin{array}{c|c|c}
21 & 6 \\
10 & 12 \\
5 & 24 \\
2 & 48 \\
1 & 96 \\
\end{array}
\hspace{1cm}
\begin{array}{c|c|c}
10101 & 110 \\
1010 & 1100 \\
101 & 11000 \\
10 & 110000 \\
1 & 1100000 \\
\end{array}
\]

\[
6 + 24 + 96 = 126
\]

(Translation: “I love binary!”)
Aside: Multiplication with Russian Peasants

\[
\begin{array}{c}
110 \\
\times \quad 10101 \\
\hline
110 \\
000 \\
11000 \\
000 \\
1100000 \\
\hline
11111110
\end{array}
\]
## Symbols ↔ Numbers ↔ Binary Strings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Number</th>
<th>Binary String</th>
</tr>
</thead>
<tbody>
<tr>
<td>' '</td>
<td>32</td>
<td>'00100000'</td>
</tr>
<tr>
<td>!</td>
<td>33</td>
<td>'00100001'</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>46</td>
<td>'00101110'</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>65</td>
<td>'01000001'</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
<td>'01000010'</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>90</td>
<td>'01011010'</td>
</tr>
<tr>
<td>a</td>
<td>97</td>
<td>'01100001'</td>
</tr>
<tr>
<td>b</td>
<td>98</td>
<td>'01100010'</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>122</td>
<td>'01111010'</td>
</tr>
</tbody>
</table>

Demo:
- `chr` and `ord`
### XOR

<table>
<thead>
<tr>
<th>A XOR B</th>
<th>A</th>
<th>B</th>
<th>A XOR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 XOR 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 XOR 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 XOR 0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 XOR 1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Associative:** \((0 \text{ XOR } 1) \text{ XOR } 0 = 0 \text{ XOR } (1 \text{ XOR } 0)\)

**Commutative:** \(0 \text{ XOR } 1 = 1 \text{ XOR } 0\)

What is \(0 \text{ XOR } 1 \text{ XOR } 1 \text{ XOR } 0 \text{ XOR } 1\)?
### XOR

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A XOR B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**One-time pad**

00111101 01111010 00010011 ...

One-time pad images from www.ranum.com and www.home.egge.net
Encoding and Decoding

Original Message: l space h ...  

Binary Version: 01001001 00100000 01101000  

One-time pad: 00111101 01111010 00010011  

XOR  

Encrypted: 01110100 01011010 01111011  

In text form: t Z {}
Encoding and Decoding

Original Message: I space h ...

Binary Version: 01001001 00100000 01101000

One-time pad: 00111101 01111010 00010011

Encrypted: 01110100 01011010 01111011

One-time pad: 00111101 01111010 00010011

Original Message! 01001001 00100000 01101000
What’s the “one-time” all about?
Using one-time pads for secret sharing...
Representing images in binary

• Quadtree compression!