Banburismus and the Story So Far

Shuttle Rescue Mission
Monday Feb 23 and Wednesday Feb 25
MEC 205 until 5:30pm
http://shuttle.cs.virginia.edu:8080/
Build and program Lego Mindstorms robot to remotely sense and navigate a barren environment and retrieve a life pod from a crater.
Exam 1 Extra Credit: either show up and watch one day or write paragraph about it

One-Slide Summary
• British codebreakers used cribs (guesses), brute force, and analysis to break the Lorenz cipher. Guessed wheel settings were likely to be correct if they resulted in a message with the right linguistic properties for German.
• If you’ve guessed the right wheel settings, two adjacent letters are more likely to be the same than they are to be different letters. Double Deltas.
• We can tell if two messages were encrypted using the same wheel settings (= same key) because the output letters will match when the input letters match. So we can try to “line them up” using Banburismus to look for matches.
• Tree sorting is only efficient if the trees are balanced. If not, it’s \( \Theta(n^2) \). The best possible sorting is \( \Theta(n \log n) \).

Breaking WWII Traffic
• Knew machine structure, but a different initial configuration was used for each message
• Need to determine wheel setting:
  - Initial position of each of the 12 wheels
  - 1271 possible starting positions
  - Needed to try them fast enough to decrypt message while it was still strategically valuable
This is what you did for PS4 (except with fewer wheels)

Outline
• WWII Codebreaking
• Double Deltas
• Machines
• Banburismus
• Tree Sorting
• Course Roadmap

Recognizing a Good Guess
• Intercepted Message (divided into 5 channels for each Baudot code bit)
  \[ Z_c = z_0 z_1 z_2 z_3 z_4 z_5 ... \]
  \[ z_{c,i} = m_{c,i} \oplus x_{c,i} \oplus s_{c,i} \]
  Message \ Key \ (parts from S-wheels and rest)
• Look for statistical properties
  - How many of the \( z_{c,i} \)’s are 0? \( \frac{1}{2} \) (not useful)
  - How many of \( (z_{c,i+1} \oplus z_{c,i}) \) are 0? \( \frac{1}{2} \)
Double Delta

\[ \Delta Z_{c,i} = Z_{c,i} \oplus Z_{c,i+1} \]

Combine two channels:

\[ \Delta Z_{1,i} \oplus \Delta Z_{2,i} = \Delta M_{1,i} \oplus \Delta M_{2,i} \]
\[ \oplus \Delta X_{1,i} \oplus \Delta X_{2,i} = \frac{1}{2} \text{ (key)} \]
\[ \oplus \Delta S_{1,i} \oplus \Delta S_{2,i} > \frac{1}{2} \text{ Yippee!} \]

Why is \( \Delta M_{1,i} \oplus \Delta M_{2,i} > \frac{1}{2} \)?

Message is in German, more likely following letter is a repetition than random

Why is \( \Delta S_{1,i} \oplus \Delta S_{2,i} > \frac{1}{2} \)?

S-wheels only turn when M-wheel is 1

Actual Advantage

- Probability of repeating letters
  \[ \text{Prob}[\Delta M_{1,i} \oplus \Delta M_{2,i} = 0] = 0.614 \]
  3.3% of German digraphs are repeating

- Probability of repeating S-keys
  \[ \text{Prob}[\Delta S_{1,i} \oplus \Delta S_{2,i} = 0] = 0.73 \]

\[ \text{Prob}[\Delta Z_{1,i} \oplus \Delta Z_{2,i} \oplus \Delta X_{1,i} \oplus \Delta X_{2,i} = 0] = 0.614 \times 0.73 + (1-0.614) \times (1-0.73) \]
\[ \Delta M \text{ and } S \text{ are 0} \]
\[ \Delta M \text{ and } S \text{ are 1} \]
\[ = 0.55 \text{ if the wheel settings guess is correct (0.5 otherwise)} \]

Using the Advantage

- If the guess of X is correct, should see higher than \( \frac{1}{2} \) of the double deltas are 0
- Try guessing different configurations to find highest number of 0 double deltas
- Problem:
  \# of double delta operations to try one config
  = length of Z * length of X
  = for 10,000 letter message = 12 M for each setting
  * 7 \( \oplus \) per double delta
  = 89 M \( \oplus \) operations (that’s a lot!)

Need a fast way to compute XOR!

Heath Robinson Machine

- Dec 1942: Decide to build a machine to do these \( \oplus \) s quickly, due June 1943
- Apr 1943: first “Heath Robinson” machine is delivered!
  - Predecessor to Colossus
- Intercepted ciphertext on tape:
  - 2000 characters per second = 30 mph
  - Perform 5 double deltas simultaneously
  - Speedup = 2.5X for faster tape * 5X for parallelism

Heath Robinson, British Cartoonist (1872-1944)

Colossus Design

- Heath Robinson machines were too slow
- Colossus designed and first built in Jan 1944
- Replaced keytext tape loop with electronic keytext generator
- Speed up ciphertext tape:
  - 5,000 chars per second = 30 mph
  - Perform 5 double deltas simultaneously
  - Speedup = 2.5X for faster tape * 5X for parallelism
Impact on WWII
- 10 Colossus machines operated at Bletchley park
- Various improvements in speed
- Decoded 63 million letters in Nazi command messages
- Learned German troop locations to plan D-Day (knew the deception was working)

Colossus History
Kept secret after the war, all machines destroyed

How could the folks at Bletchley Park solve a problem ~ 1 quintillion times harder than ps4?

Motivation Helps...
Confronted with the prospect of defeat, the Allied cryptanalysts had worked night and day to penetrate German ciphers. It would appear that fear was the main driving force, and that adversity is one of the foundations of successful codebreaking.

Simon Singh, The Code Book

Liberal Arts Trivia: Maritime Law
- A letter of marque is an official government document authorizing an agent to search, seize, or destroy specified assets or personnel belonging to a foreign party beyond the borders of the nation ("marque" or frontier). They are usually used to authorize private parties to raid and capture merchant shipping of an enemy nation. In the past, a ship operating under a letter of marque and reprisal was privately owned and was called a "private man-of-war" or ... what?
Liberal Arts Trivia: Geography

This capital city of Uttar Pradesh, the most populous state of India, is popularly known as the The City of Nawabs. It is also known as the Golden City of the East, Shiraz-i-Hind and The Constantinople of India. It is a center of Hindi and Urdu literature, and the birthplace of Kathak, a classic Indian dance form. The city was besieged during the Indian Rebellion of 1857.

Banburismus

Given two Enigma-encrypted messages, how can we determine if they were encrypted starting with the same wheel settings?

Enigma

- Invented commercially, 1923
- German Navy, Army, Air Force
- About 50,000 in use (many were captured by Allies)
- Modified throughout WWII, Germans believed perfectly secure
- Kahn’s Codebreakers (1967) didn’t know it was broken
- Turing’s 1940 Treatise on Enigma declassified in 1996

Reverse Engineering Enigma

“This fictional movie about a fictional U.S. submarine mission is followed by a mention in the end credits of those actual British missions. Oh, the British deciphered the Enigma code, too. Come to think of it, they pretty much did everything in real life that the Americans do in this movie.”

Roger Ebert’s review of U-571 (2000 Academy Award Winner)

Simple Substitution Ciphers

<table>
<thead>
<tr>
<th>ABCDEFGHIJKLMNOPQRSTUVWXYZ</th>
<th>JIDKQACRSHLGWNFEXUZVTPMYOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>encrypt</td>
<td>decrypt</td>
</tr>
<tr>
<td>HELLO ⇒ RQGGF</td>
<td></td>
</tr>
</tbody>
</table>
**Language is Non-Random**

- Random strings: the probability of two letters in the two messages matching is $1/26$ (number of letters in alphabet)

- Same-encrypted strings: the output letters will match when the input letters match
  - This happens much more frequently because some letters (e.g., “e” is ~13% of all letters) are more common

**Alan Turing’s Solution**

```
M1: GXYBGDSLWBDJLKWPEHVYGQ2WDTHRQXIKEEQ
M2: YNSCFCPVIPSMGIZEWFLHESCVIPXRMCQAXVXDU
```

**Banburismus**

```
M1: GXYBGDSLWBDJLKWPEHVYGQ2WDTHRQXIKEEQ
M2: YNSCFCPVIPSMGIZEWFLHESCVIPXRMCQAXVXDU
```
**Intercepted Message 1**

**Trying Possible Alignments**

- GXCYBGDSLWBDJLWKIFEHVQG2QWDHRQXIKEESQS
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

... YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAX...

- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

**Trying Possible Alignments**

- GXCYBGDSLWBDJLWKIFEHVQG2QWDHRQXIKEESQS
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

... YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAX...

- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

**Trying Possible Alignments**

- GXCYBGDSLWBDJLWKIFEHVQG2QWDHRQXIKEESQS
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU
- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

... YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAX...

- YNSCFCCPVIPEMSGIZWFLHESCIYSPVRXMCFOAXVXDVU

**Turing’s Hut 8 at Bletchley Park**

Don’t complain about your working space (or Small Hall). You can do good computer science anywhere. But find a quiet, undisturbed place to work on the exam.
Liberal Arts Trivia: Geology
• A stratovolcano or composite volcano is a tall, conical volcano made of many layers of lava, tephra and volcanic ash: they are characterized by steep sides and periodic eruptions. They are common in subduction zones where the ocean crust is drawn under the continental crust. Mount St. Helens and Mount Fuji are both stratovolcanos: name the country containing each one.

Liberal Arts Trivia: Mythology
• In Egyptian mythology, this falcon-headed son of Isis and Osiris fought with Seth for the throne of Egypt. In the battle his eye was wounded and later healed by Isis; this became an important symbol for renewal. He united Egypt and bestowed divinity on the pharaohs (who were viewed as his living incarnations). Name this sun, sky and war god, shown here in hieroglyphs:

The Story So Far

insert-one-tree
(define (insert-one-tree cf el tree)
  (if (null? tree)
      (make-tree null el null)
      (if (cf el (get-element tree))
         (make-tree
          (insert-one-tree cf el (get-left tree))
          (get-element tree)
          (insert-one-tree cf el (get-right tree))))
      (make-tree
       (get-left tree)
       (get-element tree)
       (insert-one-tree cf el (get-right tree))))

Each time we call insert-one-tree, the size of the tree approximately halves (if it is well balanced).

Each application is constant time.

The running time of insert-one-tree is in $\Theta(\log n)$ where $n$ is the number of elements in the input tree, which must be well-balanced.

insert-sort-helper
(define (insert-sort-helper cf lst)
  (if (null? lst) null
      (insert-one-tree cf (car lst)
         (insert-sort-helper cf (cdr lst))))

No change (other than using insert-one-tree)...but evaluates to a tree not a list!

extract-elements
(define (extract-elements tree)
  (if (null? tree) null
      (append (extract-elements (get-left tree))
       (cons (get-element tree)
         (extract-elements (get-right tree))))))

We need to make a list of all the tree elements, from left to right.

(((()) 1 ()) 2 ()) 5 () 8 ()))
Running time of insert-sort-tree

\[
\text{θ}(n \log n)
\]

\( n \) = number of elements in lst

\( n \log_2 n \)

\( n \) = number of elements in tree

\[\Theta(n \log n)\]

What if tree is not well-balanced?

A pathologically unbalanced tree is as bad as a list!

insert-one worst case requires \( n \) recursive applications, so insert-sort-tree worst case is in \( \Theta(n^2) \)

Can we do better?

• Making all those trees is a lot of work
• Can we divide the problem in two halves, without making trees?

This is the famous “Quicksort” algorithm invented by Sir Tony Hoare. See Course Book.

Computer Science: CS150 so far

• How to describe information processes by defining procedures
  - Programming with procedures, lists, recursion
    - Chapters 3, 4, 5

• How to predict properties about information processes
  - Predicting running time, \( \Theta, \Omega \)

• How to elegantly and efficiently implement information processes
  - Chapter 3 (rules of evaluation)
  - Chapter 6 (machines)

Course Roadmap

Synthesis

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Language</td>
</tr>
<tr>
<td>3</td>
<td>Programming</td>
</tr>
<tr>
<td>4</td>
<td>Procedures</td>
</tr>
<tr>
<td>5</td>
<td>Data</td>
</tr>
<tr>
<td>6</td>
<td>Machine</td>
</tr>
<tr>
<td>7</td>
<td>Cost</td>
</tr>
<tr>
<td>8</td>
<td>Time</td>
</tr>
<tr>
<td>9</td>
<td>Models</td>
</tr>
<tr>
<td>10</td>
<td>State</td>
</tr>
<tr>
<td>11</td>
<td>Objects</td>
</tr>
<tr>
<td>12</td>
<td>Computation</td>
</tr>
<tr>
<td>13</td>
<td>Algorithms</td>
</tr>
<tr>
<td>14</td>
<td>Tractability</td>
</tr>
<tr>
<td>15</td>
<td>Meta-Language</td>
</tr>
<tr>
<td>16</td>
<td>Building Web Applications</td>
</tr>
</tbody>
</table>

Analysis

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Programming</td>
</tr>
<tr>
<td>4</td>
<td>Procedures</td>
</tr>
<tr>
<td>5</td>
<td>Data</td>
</tr>
<tr>
<td>6</td>
<td>Machine</td>
</tr>
<tr>
<td>7</td>
<td>Machine</td>
</tr>
<tr>
<td>8</td>
<td>State</td>
</tr>
<tr>
<td>9</td>
<td>Object</td>
</tr>
<tr>
<td>10</td>
<td>Models</td>
</tr>
<tr>
<td>11</td>
<td>Algorithms</td>
</tr>
<tr>
<td>12</td>
<td>Computation</td>
</tr>
<tr>
<td>13</td>
<td>Tractability</td>
</tr>
<tr>
<td>14</td>
<td>Meta-Language</td>
</tr>
<tr>
<td>15</td>
<td>Building Web Applications</td>
</tr>
</tbody>
</table>
**CS150 upcoming**

- How to describe information processes by defining procedures
  - Programming with state, objects, networks
- How to predict properties about information processes
  - What is the fastest process that can solve a given problem?
  - Are there problems which can't be solved by algorithms?
- How to elegantly and efficiently implement information processes
  - How to implement a Scheme interpreter

**The Liberal Arts**

The Liberal Arts are divided into two groups: the Trivium (3 roads) and the Quadrivium (4 roads).

- **Trivium** (3 roads):
  - Language
  - Grammar
  - Rhetoric
  - Logic
- **Quadrivium** (4 roads):
  - Numbers
  - Arithmetic
  - Geometry
  - Music
  - Astronomy

**From Lecture 1:**

- Grammar: study of meaning in written expression
- Rhetoric: comprehension of verbal and written discourse
- Logic: argumentative discourse for discovering truth
- Arithmetic: understanding numbers
- Geometry: quantification of space
- Music: number in time
- Astronomy

**BNF replacement rules for describing languages, rules of evaluation for meaning**

- Not much yet… interfaces between components (PS6-9), program and user (PS8-9)
- Rules of evaluation, if, recursive definitions
- Not much yet… wait until April
- Curves as procedures, fractals (PS3)
- Yes, listen to “Hey Jude!”

**Homework**

- Exam 1 Due Wednesday Feb 25
  - Out Today

**Liberal Arts Checkup**

- Grammar: study of meaning in written expression
- Rhetoric: comprehension of verbal and written discourse
- Logic: argumentative discourse for discovering truth
- Arithmetic: understanding numbers
- Geometry: quantification of space
- Music: number in time
- Astronomy

**Read Neil deGrasse Tyson’s essay**