About This Presentation

This presentation belongs to the lecture series entitled “Ten Puzzling Problems in Computer Engineering,” devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

<table>
<thead>
<tr>
<th>Edition</th>
<th>Released</th>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>May 2007</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Word Search Puzzles
Type 1, With Word List Supplied

The puzzle below is a little harder than the normal word search: one of the 36 first/last names has been left out (which one?)

AGITATOR
ASSEMBLY
CLUTCH
CONNECTORS
CONTROL
COUPLING
GLIDE
LINT SCREEN
PULLEY
SEAL
SWITCH
VALVE

May 2007

UCSB

String Matching
“Ten Puzzling Problems in Computer Engineering”

Word Search

Word List:
- Binary Search
- Byzantine Generals
- Cryptography
- Easy Hard Impossible
- Malfunction Diagnosis
- Placement and Routing
- Satisfiability
- Sorting Networks
- String Matching
- Task Scheduling

Puzzle generated at:
http://puzzlemaker.school.discovery.com/WordSearchSetupForm.html
Word Search Puzzle
Type 2, With Clues Supplied for the Words to be Found

Seven birds

Five units of length

Four currencies

Two things football players wear

Large gland in the neck

USA Today’s “Word Roundup” for May 16, 2007: http://puzzles.usatoday.com/
String Matching: Problem Definition

Given a data string with \( n \) symbols and a pattern string with \( m \) symbols:

1. Does the pattern string appear in the data string?
2. What are the locations of all occurrences of the pattern in the data?

The brute-force, or sliding window, algorithm
Consider all possible positions where the pattern might begin \((n - m + 1)\)
For each start position, do up to \( m \) comparison to see if there is a match

Worst-case complexity = \( O(mn) \); e.g., pattern “aaaaa”, data “aaaaaaaaaaaaa”

Pattern string of length \( m = 5 \) symbols: \textbf{EAGLE}

Data string of length \( n = 96 \) symbols

\[ \text{LEAGLEUROKRDPOXWYARDRXEOIEOTHYROIDTLHNSNTETPBNEL} \]
\[ \text{AJCOZSLMOIMAFWZOHCJNMIUNRKFJERSEYELNBVEGRETXZJTED} \]
Converting 2D Search Puzzles to 1D Searches

A 2D word search puzzle looks more exotic but it can be readily converted to a 1D string search puzzle.

Row-major order

Column-major order

Similarly for (anti)diagonal

Insert a special symbol (#) between rows to ensure that new words or patterns are not created by the expansion.
Finding a Needle in a Haystack

Search for the 10-symbol “needle” h-e-l-e-n- -h-u-n-t in the Internet “haystack” with many TBs of data

The brute force algorithm amounts to the following: “Look in this corner, now in this other corner, then over there, and so on.”

The Internet holds some 20B pages, each page on average containing in excess of 100 KB

\[ m \approx 10 \]
\[ n \approx 20 \times 10^9 \times 10^5 = 2 \times 10^{12} \text{ B} \]
\[ O(mn) \approx 2 \times 10^{13} \text{ comparisons} \Rightarrow 20,000 \text{ s (} > 5 \text{ hr)} \text{, with } 10^9 \text{ comparisons/s} \]
Needle in a Haystack: Internet Search

Search for the 10-symbol string h-e-l-e-n- -h-u-n-t
Needle in a Haystack: Doing Less Work

For a particular pattern and unpredictable data strings, preprocess the pattern so that searching for it in various data strings becomes faster.

Analogy: Magnetize the needle

For a particular data string and unpredictable patterns, preprocess the data string so that when a pattern is supplied, we can readily find it with much less work.

Analogy: Do a thorough search of the haystack for different types of needles and place markers to guide future searches.
Example of Preprocessing the Pattern String

Devise an efficient method for finding the pattern “abcbab” in various data strings formed from the symbols a, b and c.

Data string: a b c b b b a b c b a b b c a a b c b a b c b b

O(n) instead of O(mn)
Example of Preprocessing the Data String

Devise an efficient method for finding various patterns in the data string:

```
  a b c b b b a b c b a b b c a a b c b a b c b a b c b b
```

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b b</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>a b c</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>b c a</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>b b b</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>b c b</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>c b a</td>
<td>0, 6, 15, 19, 23</td>
</tr>
<tr>
<td>b a b</td>
<td>0, 6, 15, 19, 23</td>
</tr>
</tbody>
</table>

Find all occurrences of the pattern “abcbab”

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b c</td>
<td>5, 9, 18, 22</td>
</tr>
<tr>
<td>b c b</td>
<td>5, 9, 18, 22</td>
</tr>
<tr>
<td>c b a</td>
<td>5, 9, 18, 22</td>
</tr>
<tr>
<td>b a b</td>
<td>5, 9, 18, 22</td>
</tr>
</tbody>
</table>

Alternate strategy: Focus on the locations of a b c and b a b
Search Engine Indexes

- Google Search
  - Search for "the"
  - Results: 50 of about 5,290,000,000 for the (0.15 seconds)

- The Onion - America's Finest News Source
  - Welcome to the White House
  - Whitehouse.gov is the official web site for the White House and President George W. Bush, the 43rd President of the United States.

- Search for "xmzt"
  - Results: 50 of about 667 for xmzt (0.10 seconds)

Did you mean: "kmzt"
Approximate String Matching

Notion of string distance
Each of the following transformations in a string creates a distance of 1

1. Deletion of a symbol
2. Insertion of an extra symbol
3. Transposition of two adjacent symbols

Example distance-1 strings for helen hunt:

- hellen hunt
- elen hunt
- helen hnut

Example distance-2 strings for helen hunt:

- hellen hnut
- elen hnut
- lheen hunt

Wildcard symbols can help in formulating approximate string searches

h* hunt means any string that begins with an “h”, ends with “hunt”, and has an arbitrary set of symbols between the two

Melvyl (UC library catalog) allows such searches, e.g., author: hunt h*