All Permutations Considered

Andrew Mertz
William Slough
Nancy Van Cleave

Mathematics and Computer Science Department
Eastern Illinois University

September 24, 2005
Overview

- National Public Radio *Weekend Edition* Sunday Puzzle
- A few puzzles
- Common thread – permutations
- Making permutation generation accessible
- Student activities
- Classroom discussion
- Why is this nifty?
NPR Weekend Edition Quiz

- Weekend Edition Sunday Puzzle with Will Shortz
- Variety of word and number puzzles
- http://www.npr.org/programs/wesun/puzzle
- Why puzzles?
  - General interest and appeal
  - No special background needed
  - Many connections to computing
Sample Problem

<table>
<thead>
<tr>
<th>T</th>
<th>E</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>O</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>R</td>
<td>Y</td>
</tr>
</tbody>
</table>

Diagram 1:

Diagram 2:

Diagram 3:
Another Problem
Another Problem

War
Another Problem

War Wag
Another Problem

War Wag Wig
Another Problem

War Wag Wig Wit
ANALOGIESK
The Common Thread: Generating Permutations

- Recursive?
- Iterative?
- Interface?
- Implementation?
- All permutations or one-at-a-time?
Making Permutations Accessible: Usage

...  
PermutationChain MyChain("abc");

for (int i = 0; i < 6; i++)
    cout << MyChain.Draw() << endl;
...

CCSC:MW 2005
Making Permutations Accessible: Interface

#include <vector>
#include <string>

class PermutationChain {
    public:
        // constructor
        PermutationChain(const string & s);

        // facilitator
        string Draw();

    private:
        vector<int> Current;
        string TheString;
        bool IsOrdered();
};
Part 1. Generate all permutations of an input string

Part 2. Given a string of size 9, generate all ten 3-letter words from the grid

Part 3. Construct a vector of strings and determine whether they’re all in the dictionary

Part 4. Using PermutationChain and the two functions implemented in parts 1–3, solve the puzzle
Classroom Discussion

- Interface vs. implementation — focus on interface
- Seat-of-the-pants analysis
  - Linear search vs. binary search
  - Generalization: $4 \times 4$ grid
  - Space considerations
  - Estimations of time needed
- JUMBLE puzzle
- What other 9-letter words have this property?
Why is this Nifty?

► It’s fun!
► It’s interesting!
► Surprises for students:
  ► Wow; I can actually make a computer do this?
  ► Not all programs run in less than one second!
  ► How many seconds (days? months? years?) for a $4 \times 4$?!
► Provides springboard to topics often not seen until later
  ► Permutation generation
  ► Analysis of algorithms — both time and space
  ► Feasible computations