Algorithmic Problem Solving

IS 101Y/CMSC 104Y
First Year IT

Penny Rheingans
University of Maryland Baltimore County

• Questions
• Announcements
• Names
Important Problems

- Health
  - Cure disease, increase health coverage,
- Prosperity
  - End poverty, restore world economy, end world hunger, improve safety and cybersecurity, spread education, create job opportunities, decrease gas prices
- Environment
  - Manage natural resources, reduce pollution, stop global warming
- Scientific/technical discovery
  - Develop hovercars, explore space
- Freedom/Justice
  - Increase equality, stand against oppression, reduce partisanship, establish world peace, fix foreign policy,
- Personal fulfillment
  - Be happy, build good relationships, learn from mistakes, express self, be financially independent, take risks/chances, spread love, improve time management,

A Kidney Story

- Kidney disease affects 50,000 new Americans a year
- Transplants as treatment
  - Pairs
  - Cycles
  - Chains

Two cycle

Three cycle
A Big Kidney Story

What about really big chains?

How do you come up with optimal series of swaps?

For more info:

Kidney Exchange

• Consider the exchange below. A patient is connected to a donor if they are biologically compatible. A donor will only donate a kidney if his or her friend also receives a kidney. What is the optimal matching for this exchange? Why?

• What technique did you use to solve this problem? How would your technique scale if there were ten donors and patients? 100? Thousands?
Alternative Representation

- A graph data structure can capture the important relationships among patients and donors.

![Graph diagram]

- A legal exchange is one where there is a path following edges that visits each vertex exactly once and returns to the starting vertex (if every node = Hamiltonian cycle, http://nrich.maths.org/2320).
- What is the longest such path in this graph?

Algorithms

- An algorithm is an ordered set of unambiguous steps that describes a process.
- Examples from real life
Examples

Gingerbread Baby

Ingredients:
- 1 1/2 cups all-purpose flour
- 1 teaspoon baking soda
- 1/2 teaspoon salt
- 1 1/2 cups sugar
- 1 cup unsalted butter, softened
- 1 cup molasses
- 1 egg
- 2 teaspoons vanilla extract

Instructions:
1. Preheat oven to 350°F.
2. In a large mixing bowl, cream together the butter and sugar.
3. Add the molasses and egg.
4. In a separate bowl, mix the flour, baking soda, and salt.
5. Gradually add the flour mixture to the butter mixture.
6. Blend in the vanilla extract.
7. Roll out the dough to 1/4 inch thickness.
8. Cut out shapes with cookie cutters.
9. Place on a baking sheet lined with parchment paper.
10. Bake for 8-10 minutes or until edges are firm.
11. Let cool on the baking sheet.

Makes 1 dozen cookies.
Driving directions to 1501 W Covell Rd, Edmond, Oklahoma, 73013

Suggested route:

1. US-77 N 12.3 mi 22 min
2. N Pennsylvania Ave 25 min
3. S Boulevard St 24 min

1. Head west on W Wilshire Blvd toward Nichols Rd
2. Turn left at Broadway Extension Service Rd
3. Take the ramp to the left onto US-77 N
4. Take the Memorial Rd W exit
5. Merge onto NE 159th Blvd Memorial Rd
6. Turn right at N Kelly Ave

Finish parking Edmond OK

Fun Cello

L. van Beethoven (1770-1827)
Friendship

- Friendship Algorithm

Birthdays

- In your group, sort members by increasing order of birthday.

- Sorting algorithms
Whipped Butter Frosting

7 1/2 T all purpose flour
1 1/2 C milk
1 1/2 C butter
1 1/2 C granulated sugar
3/8 t salt
3 t vanilla


Whipped Butter Frosting: Revisited

7 1/2 T all purpose flour
1 1/2 C milk
1 1/2 C butter
1 1/2 C granulated sugar
3/8 t salt
3 t vanilla

Pseudocode

• Specialized language for describing algorithms
  – General syntax and semantics
  – Not specific to a particular computer language

• Some syntax
  <THING> - any item that is a THING
  { ... } - stuff between braces grouped

Pseudocode Primitives

• Procedures and functions
  procedure <NAME> (<VAR>, ...) { ... }
  function <NAME> (<VAR>, ...) { ... return <EXPR>; }

• Assignments and printing
  <VAR> = <EXPR> - assignment
  print <EXPR>, ...

• Mathematical Expressions <EXPR>
  <VAR> - variable value
  <EXPR> + <EXPR> - result of operation
  "<any string>" - string
  <LIST>[<I>] - the Ith item in a list

• Logical Conditions <COND>
  <EXPR1> = = <EXPR2> - test for equality
  <EXPR1> < <EXPR2> - test for inequality
  not <COND> - test for state
More Pseudocode Primitives

• Conditional logic
  if <CONDITION> then { ... }
  if <CONDITION> then { ... } else { ... }

• Loops
  for <ITEM> in <LIST> do { ... }
  for <I> from <MIN> to <MAX> do { ... }
  while <CONDITION> do { ... }
  do { ... } until <CONDITION>

Example: Counting

// Print the numbers from 1 to N
procedure PrintNumbers (N) {
  for i from 1 to N {
    print (i)
  }
}

Example: Order

- Given the following sentences:
  1. I don’t want pizza again for a long time.
  2. I ate ten pieces of pizza.
  3. Later that night, I got sick.
  4. I felt very full.
- Which of the following orders is correct?
  A) 1, 3, 4, 2
  B) 4, 3, 2, 1
  C) 2, 3, 1, 4
  D) 3, 1, 4, 2
  E) 2, 4, 3, 1
- Concepts: state, search space

Exercise: Guessing

- With team, write pseudocode for guessing a number between 1 and 100. Be prepared why your approach is the best way to solve the problem.
Example: Multiplication

How would you multiply two numbers, using only the addition operator?

Concepts: iterations, efficiency

Exercise: Socks

• With team, write pseudocode for sorting a pile of socks. Be prepared why your approach the best way to solve the problem.
Algorithms are used to express solutions to computational problems.

• An algorithm is a precise sequence of instructions for a process that can be executed by a computer.
  – Sequencing, selection, iteration, and recursion are building blocks.
  – Different algorithms can be developed to solve the same problem.

• Algorithms are expressed and implemented using languages.
  – natural language, pseudo-code, and visual and textual languages.
  – better suited for expressing different algorithms.
  – can affect clarity or readability, but not whether solution exists.

• Algorithms can solve many, but not all, problems.
  – Many problems can be solved in a reasonable time.
  – Some need heuristic approaches to solve them in a reasonable time.
  – Some problems cannot be solved using any algorithm.

• Algorithms are evaluated analytically and empirically.
  – using many criteria (e.g., efficiency, correctness, and clarity).
  – algorithms for the same problem can have different efficiencies.