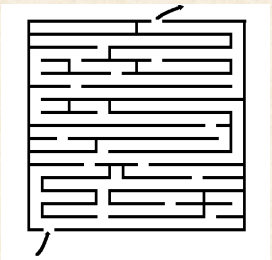
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202: Introduction to Computation — Professor Andrea Arpaci-Dusseau

What is an Algorithm?



What is Computer Science?

Computer science studies algorithms

1. Formal and mathematical properties
2. Hardware realizations
3. Linguistic realizations
4. Applications

What is an algorithm?

- Procedure for solving a problem in finite number of steps
- Step-by-step method for accomplishing a task

Notice any words that are not in that definition?

- Does not mention "with a computer" or "with particular programming language"

Algorithm Example: Recipe for Brownies

½ cup butter or margarine
1 cup sugar
2 eggs
1 tsp vanilla extract
½ cup cocoa
½ cup flour

1. If butter not soft, then melt butter
2. Blend melted butter and sugar until mixture has creamy consistency
3. Add eggs and vanilla; stir
4. Add cocoa and flour; mix until well blended
5. Pour into greased round glass cake pan
6. Microwave for 8-9 minutes

Algorithm Example: Taxes

Line 1: Write your total wages from your W-2s
Line 2: Write your total interest from your 1099-INTs
Line 3: Write your unemployment compensation from 1099-Gs

- If you received Alaska Permanent Fund dividends only, then put the figure reported by the State of Alaska on Line 3
- If unemployment and Alaskan dividends, then put total on Line 3

Line 4: Add lines 1, 2, 3 for your Adjusted Gross Income (AGI)
Line 5: Determine your personal exemptions

- If being claimed as a dependent, then check "Yes"; else check "No"
- If you are unmarried, or you are married and you are not filing a joint return, then write $7,950; otherwise, write $15,900

Line 6: Subtract line 5 from line 4 for your taxable income

Algorithm Example: Knitting a Scarf

1. Hold needle with stitches in left hand; insert point of right needle in first stitch
2. With right index finger, bring yarn under and over right needle
3. Draw yarn thru stitch with right needle point
4. Slip loop on left needle off, so stitch is on right needle
5. This completes one stitch. Repeat Steps 1-4 in each stitch still on left needle. When the the last stitch is worked, one row is done and you can move to Step 6
6. Measure your work. It should be 7" wide. If it is too wide, start over and cast on fewer stitches; if it is too narrow, start over and cast on more stitches

Algorithms

Properties

- Take set of inputs to operate over
- Has precise steps that express operations to perform
  - Assume some basic set of known primitive operations
- Produces some output

Control Flow

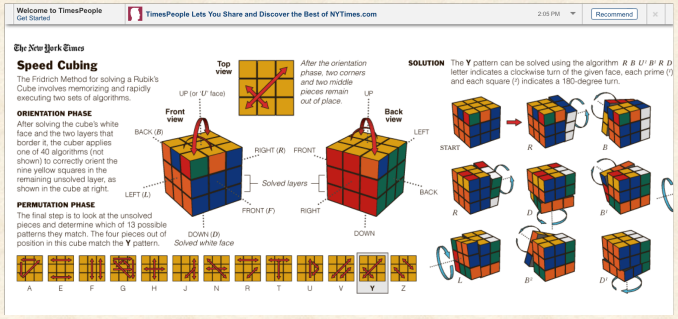
- Default: execute instructions sequentially (in order)
- May alter control flow with "if…then" or "repeat"
- May ask questions using logical "or" or "and"
- May define new operations (e.g., "stitch")

Other Example Algorithms?

Where do you use algorithms in your life?

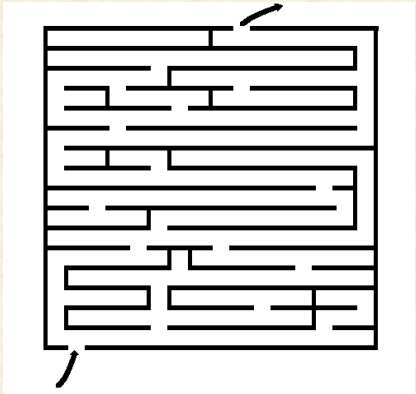
- Putting together IKEA furniture
- Folding paper airplanes
- Looking up a word in the dictionary
- Getting home from school
- Solving a jigsaw puzzle
- Solving a sudoku puzzle
- Playing tic-tac-toe
- Playing chess
- Solving rubik's cube

Algorithm Example: Solving a Rubik's Cube



Robots can even solve rubik's cubes

Exercise: How do you solve a maze?



What to Think About when Solving Maze?

Getting a solution to this maze is not the point

Can you figure out HOW to solve ANY maze?

Can you TELL someone else how to solve any maze?

Assume person knows certain operations:

- Draw a path with a pencil
- Can draw straight, turn left, turn right, turn around
- Detect walls, dead ends, and junctions
- Detect where you've drawn a line

What Approaches Did You Find?

What is the easiest algorithm to specify?

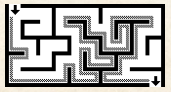
Approach #1: Random-walk

- At Junction what should you do?
  - Pick direction at random
- At dead-end what to do?
  - Just turn around
- Advantage?
  - Simple to specify and implement
  - No "state" to remember (don't look at drawn lines)
- Disadvantage?
  - Probabilistic: May take long time!

Maze Solving Approach #2

Follow-right-wall (Variant: Follow-left-wall)

- At Junction: Always go to right
  - Dead-end: Turn around w/ wall
  - Video



Advantages:

- Probably faster than random
- No state to remember

Disadvantages:

- Does not work on all mazes

Can you think of example mazes that don't work??

- Can get stuck in a cycle (island)
- Does not work if goal is not on outside edge

Maze Solving Approach #3

Depth-first search (Tremaux's algorithm)

- Draw line to record path (remember where you've been)
- At junction: if unmarked path exists, take it
  - Else, take path marked once
    - What does this correspond to?
  - Never take path marked twice
    - Why not? Why guaranteed don't ever have to?
- At dead-end, turn around
- Advantages
  - Works on all mazes
- Disadvantages
  - Must record decision at each branch

Maze Solving Approach #4

Dead-end filling

- Identify all dead-ends
- "Fill in" dead-end paths until first junction
- Avoid all dead-end paths – path to goal is left!
  - Some complications with loops
- Video

Advantages

- No back-tracking needed
- Can find goal more rapidly

Disadvantages

- Requires global view of maze

Robots can Solve Mazes

Algorithms exist for solving mazes

Maze robot competition

- Micromouse

Goal in center of maze, not on outer wall

- Phase 1: Explore maze, build internal representation
- Phase 2: Timed from start to finish

Lots of challenges other than pure maze solving

- Sensing walls, Mechanics of moving, turning robot quickly

Video

Maze: Take-Away Lesson

Numerous algorithms exist for solving mazes

Some algorithms (solutions) are better than others

What's the criteria of a good algorithm?

- Correct: Works on all mazes
  - Test by running implementation on all possible input mazes
  - Prove correct given certain assumptions
- Fast:
  - Benchmark by running implementation on particular machine for particular maze input
  - Count number of operations as function of maze size
- Minimize amount of "state" (memory) to record

Today's Check-Up

What is an algorithm?

- Step-by-step method for accomplishing a task

Which of the following are examples of algorithms?

- a recipe
- a maze
- a solution for solving a rubix cube
- a zipcode
- programming languages such as Scratch

True or False:

- An algorithm can return different output in different circumstances?

Announcements

Homework 1 Due Friday by 5pm Today

- Lab hours today in 1370 from 1:30-3:00 if still needed

Homework 2 Available Friday

- Use Scratch to create!

Tuesday is BYOL (laptop) day

- Before lecture: Get Scratch 1.4 from http://scratch.mit.edu
- Some laptops with Scratch available to borrow