What can a computer be commanded to do?

Doing What You're Told

Lawrence Snyder
University of Washington, Seattle

Announcements

- The Midterm is Monday –
  - One sheet (8.5" x 11") of notes, handwritten OK
  - No other materials except pencil & eraser – no phone, calculator, computer, books, etc.
  - The test is on paper … this means that program text – int, for, mouseX – will not be highlighted

Thinking About Computing

- Computers seem to run really fast … except when they don't
  - Usually we don't know why
  - Often it is communications congestion on I'net
  - Other times, when saving files, say, we're waiting for the hard disk to copy everything
- Often the time a computer takes to solve a problem is proportional to how much data there is … more pixels, more time to process

Time Proportional To n

- CS folks say that problems whose work is proportional to n are n-time or linear time
  - Making an image lighter in your photo software
  - Adding a column of numbers in a spreadsheet
  - Crawling the Internet looking for links
  - … many more … linear problems are common

- Apparently some problems are not …

Sorting

- Putting a sequence of items into alphabetical or numerical order

walrus seal whale gull clam

Algorithm: compare to all following items , reorder if needed

wa se wh g c
se wa wh g c
se wa wh g c
g wa wh se c
c wa wh se g
c wa wh se g
c se wh wa g
c g wh wa se
c g wa wh se
c g se wh wa
c g se wa wh

- Other ways to sort

How Long To Sort w/Exchange?

- The pattern is, for n items
  - n-1 focus on first time
  - n-2 focus on second item
  - n-3 focus on third item
  - …
  - 1 on next to last

n-1 rows in list, ave of each row is n/2, so $(n-1)\times n/2$

$= (n^2 - n)/2$

- Time proportional to n^2

wa se wh g c
se wa wh g c
se wa wh g c
g wa wh se c
c wa wh se g
c wa wh se g
c se wh wa g
c g wh wa se
c g wa wh se
c g se wh wa
c g se wa wh

Polynomial

- Other computations have running time
  - proportional to n3 – matrix multiplication
  - proportional to n4
  - …
- All of them are lumped together as "*polynomial* time computations"
  - Considered to be realistic … a person can wait
  - Polynomial, but not linear … get a computer person to help develop your solution

To Infinity And Beyond

There are more complex computations …

- Suppose you want to visit 28 cities in the US (for a rock concert?) and you want to minimize your how much you pay for airplane tickets
- You could select an ordering of cities (SEA → PDX → SFO → LAX …) and compute the ticket price.
- Then pick another ordering (SEA → SFO → LAX → PDX …), compute this ticket price and compare to the previous one
- Always keep the cheapest itinerary

This seems very dumb … isn't their a better way?

Traveling Salesman Problem

- Actually, no one knows a way to solve this problem significantly faster than checking all routes and picking the cheapest …
- Not polynomial time … guessing no poly sol'n
- This is can NP-Complete problem
  - Many many related problems … the best solution is "generate and check"
    - Best way to pack a container ship
    - Most efficient scheduling for high school students' classes
    - Least fuel to deliver UPS packages in Washington
    - Fewest public alert broadcast stations for US

Astonishing Fact

- Although there are thousands of NP-Hard Problems, meaning they're basically "generate and check" …
- NP-Complete computations (like traveling salesman) have the property that if any one of them can be done fast (n^x-time, say) the EVERYONE of the related problems can be too!

- Is Traveling Salesman solvable in n^x time is one of the great open questions in computer science

Be Famous … Answer This Question

There's Stuff A Computer Can't Do

- Some problems are too big – combinatorial explosive – like checking each chess game to see if there is a guaranteed win for White
  - Too many items to check
  - Doable in principle, however

Some Well Formed Problems

- One problem that has a clear specification but can't be solved is

Halting Problem

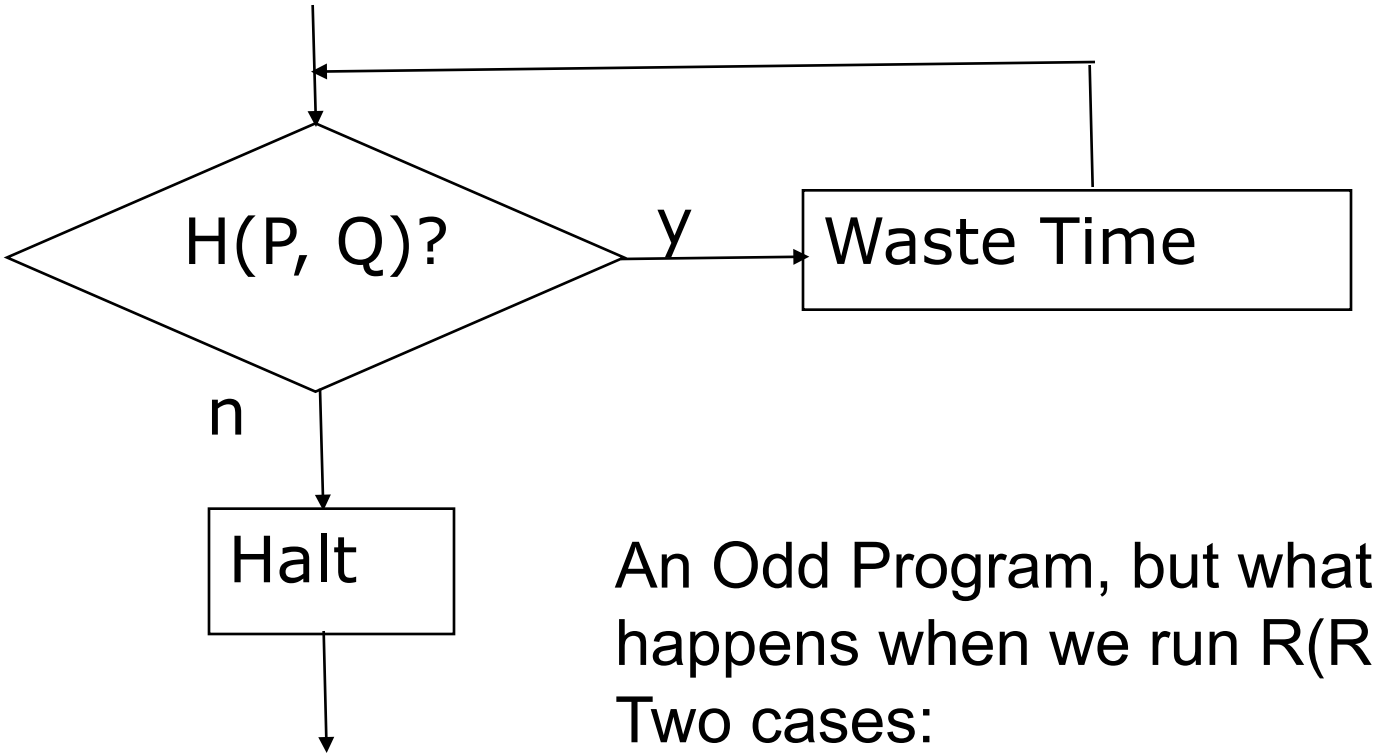
Decide, given a program P and input Q whether P(Q), that is, P run on input Q, will eventually stop running and give an answer

- This seems pretty easy … though running it won't work because it might not stop … but maybe analysis could find any errors

NOT

- The halting problem cannot be solved
- Here's why …
  - Suppose (for purposes of contradiction) that some program H(P, Q) answers the halting problem (will it halt and give an answer) for program P on data Q
  - Notice the question is not, does it give the RIGHT answer … just will it give any answer

A New Program R(P, Q)



An Odd Program, but what happens when we run R(R, R)

Two cases:

- ✗ H(R,R) says yes, it halts
- ✗ H(R,R) says no, it doesn't halt

Therefore, H cannot exist!

Summary

- We have analyzed the complexity of computations, and learned …
  - Many computations have *time proportional to n*
  - Many, like sort, have running *time proportional to* n^2
  - Others have running time proportional to n^3, n^4, …
  - Some computations are computable in principle but not in practice: *NP-complete*
  - Some things cannot be computed at all, such as the *Halting Problem*