CS10: The Beauty and Joy of Computing

Lecture #22
Limits of Computing

UC Berkeley EECS
Summer Instructor
Ben Chun

2012-07-26

You'll have the opportunity for extra credit on your project! After you submit it, you can make a ≤ 5min YouTube video.

MOUNTAIN LION ON CAMPUS?

Warning sign posted at Stern Hall. Also, Apple releases new operating system.

http://www.dailycal.org/2012/07/25/mountain-lion-sighted-near-uc-berkeley-stern-hall/

---

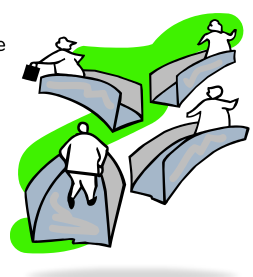
www.eecs.berkeley.edu/Research/Areas/

Computer Science … A UCB view

- CS research areas:
  - Artificial Intelligence
  - Biosystems & Computational Biology
  - Database Management Systems
  - Graphics
  - Human-Computer Interaction
  - Networking
  - Programming Systems
  - Scientific Computing
  - Security
  - Systems
  - Theory
    - Complexity theory
  - …

Let's revisit algorithm complexity

- Problems that…
  - are tractable with efficient solutions in reasonable time
  - are intractable
  - are solvable approximately, not optimally
  - have no known efficient solution
  - are not solvable



---

en.wikipedia.org/wiki/P_(complexity)

Tractable with efficient sols in reas time

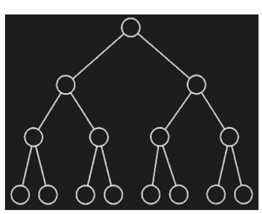
- Recall our algorithm complexity lecture, we've got several common orders of growth
  - Constant
  - Logarithmic
  - Linear
  - Quadratic
  - Cubic
  - Exponential
- Order of growth is polynomial in the size of the problem
- E.g.,
  - Searching for an item in a collection
  - Sorting a collection
  - Finding if two numbers in a collection are same
- These problems are called being "in P" (for polynomial)

---

en.wikipedia.org/wiki/Intractability_(complexity)#Intractability

Intractable problems

- Problems that can be solved, but not solved fast enough
- This includes exponential problems
  - E.g., $f(n) = 2^n$
    - as in the image to the right
- This also includes poly-time algorithm with a huge exponent
  - E.g, $f(n) = n^{10}$
- Only solve for small n



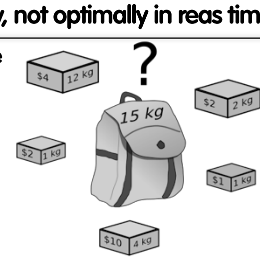
Imagine a program that calculated something important at each of the bottom circles. This tree has height n, but there are 2^n bottom circles!

---

en.wikipedia.org/wiki/Knapsack_problem

Solvable approximately, not optimally in reas time

- A problem might have an optimal solution that cannot be solved in reasonable time
- BUT if you don't need to know the perfect solution, there might exist algorithms which could give pretty good answers in reasonable time

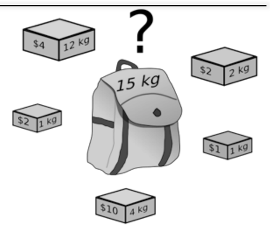


Knapsack Problem
You have a backpack with a weight limit (here 15kg), which boxes (with weights and values) should be taken to maximize value?

Peer Instruction

What's the most you can put in your knapsack?

a) $10
b) $15
c) $33
d) $36
e) $40



Knapsack Problem
You have a backpack with a weight limit (here **15kg**), which boxes (with weights and values) should be taken to maximize value?
(any # of each box is available)

Have no known efficient solution

- **Solving one of them would solve an entire class of them!**
  - We can transform one to another, i.e., reduce
  - A problem P is "hard" for a class C if <u>every</u> element of C can be "reduced" to P
- **If you're "in NP" and "NP-hard", then you're "NP-complete"**

-2 -3 15
14 7 -10

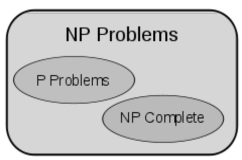
Subset Sum Problem
Are there a handful of these numbers (at least 1) that add together to get 0?

- **If you guess an answer, can I <u>verify it</u> in polynomial time?**
  - Called being "in NP"
  - Non-deterministic (the "guess" part) Polynomial

The fundamental question. Is P = NP?

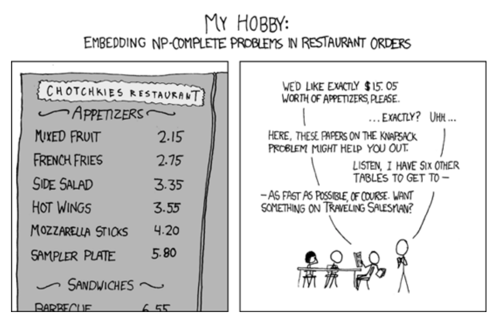
- **This is THE major unsolved problem in Computer Science!**
  - One of 7 "millennium prizes" w/a $1M reward
- **All it would take is solving ONE problem in the NP-complete set in polynomial time!!**
  - Huge ramifications for cryptography, others

If P ≠ NP, then



- **Other NP-Complete**
  - Traveling salesman who needs most efficient route to visit all cities and return home

xkcd.com/287



Problems NOT solvable

- **<u>Decision problems</u> answer YES or NO for an infinite # of inputs**
  - E.g., is N prime?
  - E.g., is sentence S grammatically correct?
- **An algorithm is a <u>solution</u> if it correctly answers YES/NO in a finite amount of time**
- **A problem is <u>decidable</u> if it has a solution**

Alan Turing
asked, "Are all problems decidable?"
(People believed yes at the time.)
Turing proved they are not!

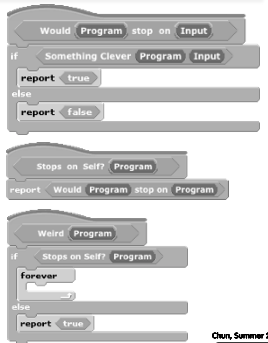
Review: Proof by Contradiction

- **Infinitely Many Primes?**
- **Assume the contrary, then prove that it's impossible**
  - Only a finite # of primes
  - Number them $p_1, p_2, \ldots, p_n$
  - Consider the number q
    - $q = (p_1 * p_2 * \ldots * p_n) + 1$
    - Dividing q by any prime would give a remainder of 1
    - So q isn't composite, q is prime
    - But we said p_n was the biggest, and q is bigger than p_n
  - So there IS no biggest p_n

Euclid
www.hisschemoller.com/wp-content/uploads/2011/01/euclides.jpg

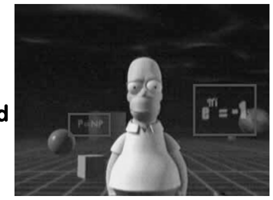
Turing's proof : The Halting Problem

- **Given a program and some input, will that program eventually stop? (or will it loop)**
- **Assume we could write it, then let's prove a contradiction**
  - 1. write Stops on Self?
  - 2. Write Weird
  - 3. Call Weird on itself



Conclusion

- **Complexity theory important part of CS**
- **If given a hard problem, rather than try to solve it yourself, see if others have tried similar problems**
- **If you don't need an exact solution, many approximation algorithms help**
- **Some not solvable!**



P=NP question even made its way into popular culture, here shown in the Simpsons 3D episode!