Class 14: Intractable Problems

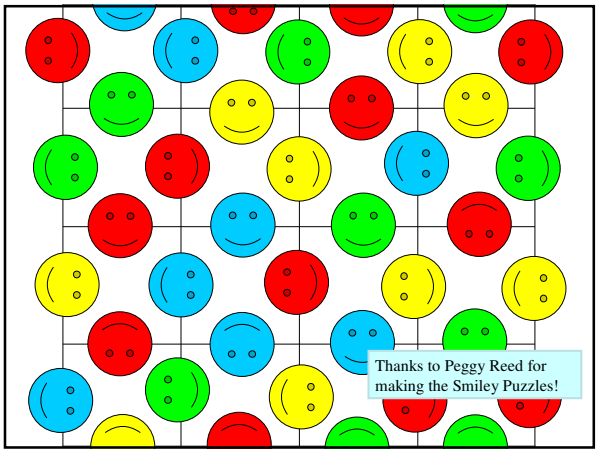
CS150: Computer Science
University of Virginia
Computer Science

David Evans
http://www.cs.virginia.edu/evans

Smileys Problem

Input: n square tiles

Output: Arrangement of the tiles in a square, where the colors and shapes match up, or "no, its impossible".



Problems and Procedures

- To know a $O(f)$ bound for a problem, we need to find a $\Theta(f)$ *procedure* that solves it
  - The sorting **problem** is $O(n \log n)$ since we know a **procedure** that solves it in $\Theta(n \log n)$
- To know a $\Omega(f)$ bound for a **problem**, we need to prove that there is no procedure faster than $\Theta(f)$ that solves it
  - We proved sorting is $\Omega(n \log n)$ by reasoning about the number of decisions needed

How much work is the Smiley's Problem?

- Upper bound: (O)
  $O(n!)$
  Try all possible permutations
- Lower bound: (Ω)
  $\Omega(n)$
  Must at least look at every tile
- Tight bound: (θ)

No one knows!

Complexity Class P "Tractable"

Class P: problems that can be solved in polynomial time
$O(n^k)$ for some constant k.

Easy problems like sorting, making a photomosaic using duplicate tiles, simulating the universe are all in **P**.

Complexity Class NP

Class NP: problems that can be solved in *nondeterministic* polynomial time

If we could try all possible solutions at once, we could identify the solution in polynomial time.

Alternately: If we had a magic guess-correctly procedure that makes every decision correctly, we could devise a procedure that solves the problem in polynomial time.

Note: this definition is not precise enough to be satisfying yet! We will need to understand better what a "step" means when we measure work to define this properly.

NP Problems

- Can be solved by just trying all possible answers until we find one that is right
- Easy to quickly check if an answer is right
  - Checking an answer is in **P**
- The smileys problem is in NP

  We can easily try $n!$ different answers

  We can quickly check if a guess is correct (check all n tiles)

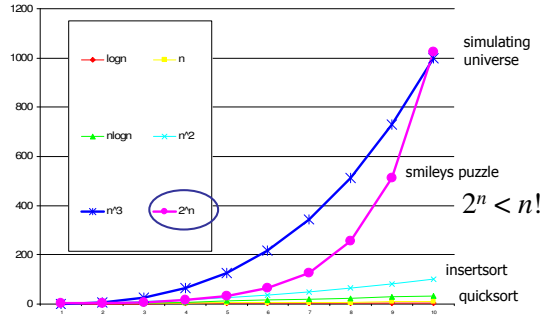
Is the Smiley's Problem in P?

No one knows!

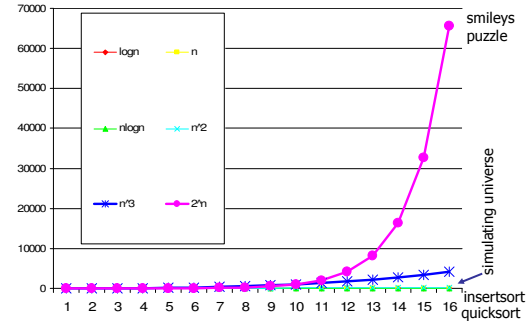
We can't find a $O(n^k)$ solution.

We can't prove one doesn't exist.

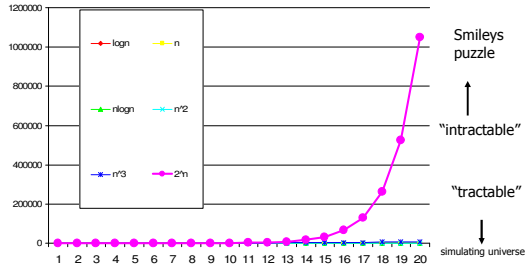
Orders of Growth



Orders of Growth



Orders of Growth

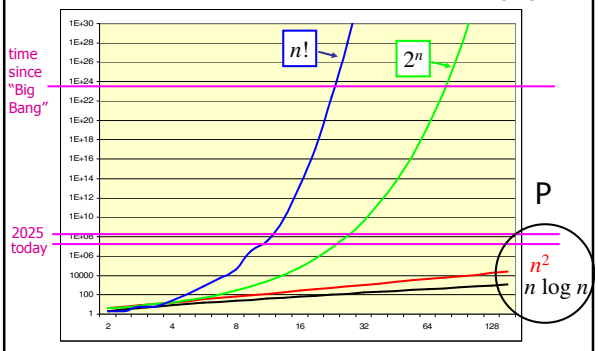


I do nothing that a man of unlimited funds, superb physical endurance, and maximum scientific knowledge could not do.
– Batman (may be able to solve intractable problems, but computer scientists can only solve tractable ones for large n)

Quiz Break

Intractable Problems

log-log scale



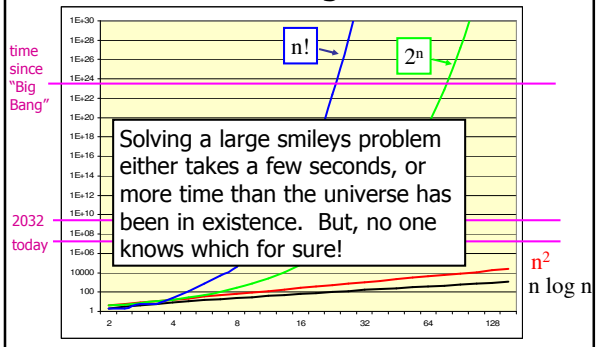
Moore's Law Doesn't Help

- If the fastest procedure to solve a problem is $\Theta(2^n)$ or worse, Moore's Law doesn't help much.
- Every doubling in computing power increases the problem size by 1.

P = NP?

- Is there a polynomial-time solution to the "hardest" problems in NP?
- No one knows the answer!
- The most famous unsolved problem in computer science and math
- Listed first on Millennium Prize Problems
  - win $1M if you can solve it
  - (also an automatic A+ in this course)

This makes a huge difference!



Charge

- PS4 is due Monday
- More on P vs. NP next class:
  - What is the hardest problem in NP?
  - What would it mean to find a fast procedure that solves the Smiley puzzle?
  - Is it ever useful to know a problem is *hard*?