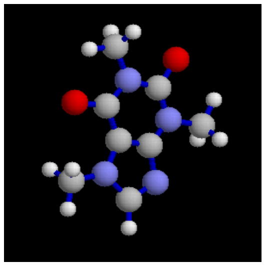
Class 15: P vs. NP (Smiley Puzzles and Curing Cancer)



CS150: Computer Science
University of Virginia
Computer Science

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

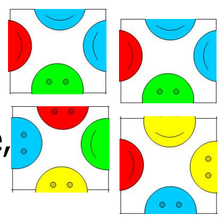
Menu

- Complexity Classes P and NP
- Quiz Answers
- Problem Reductions

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".

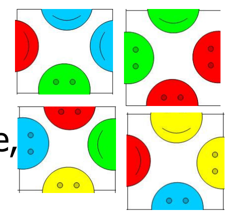


"Not possible"

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".



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!

NP Problems

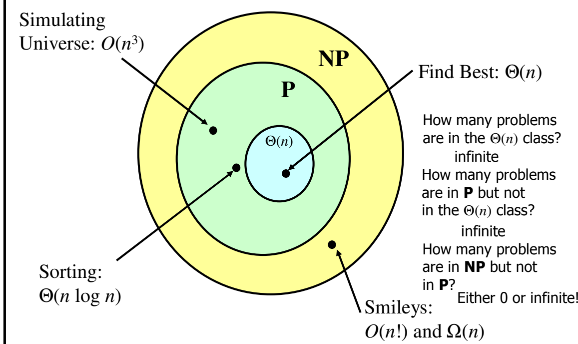
- Can be solved by just trying all possible answers until we find one that is right
- Easy to quickly check if an answer is right
  - Checking an answer is in **P**
- The smileys problem is in **NP**
  - We can easily try $n!$ different answers
  - We can quickly check if a guess is correct (check all n tiles)

Complexity Classes

Class P: problems that can be solved in polynomial time ($O(n^k)$ for some constant k): Easy problems like simulating the universe are all in **P**.

Class NP: problems that can be solved in polynomial time by a nondeterministic machine: includes all problems in **P** and some problems possibly outside **P** like the Smileys puzzle

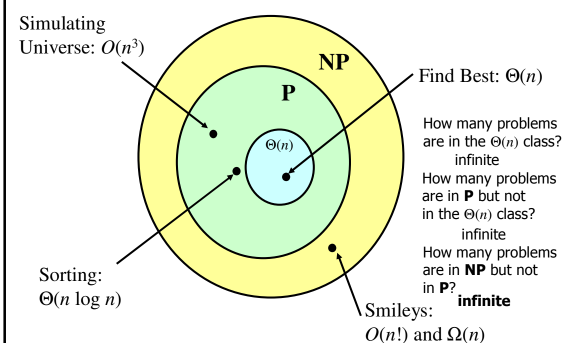
Problem Classes: Possible View



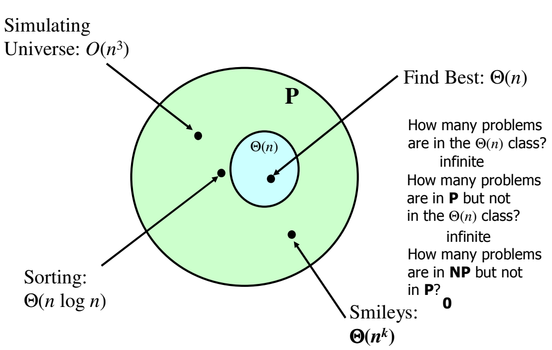
P = NP?

- Is P different from NP: is there a problem in NP that is not also in P
  - If there is one, there are infinitely many
- Is the "hardest" problem in NP also in P
  - If it is, then every problem in NP is also in P
- No one knows the answer!
- The most famous unsolved problem in computer science and math
  - Listed first on Millennium Prize Problems

Problem Classes if P ≠ NP:



Problem Classes if P = NP:



Quiz Responses

- Partners for PS3
  - Only 4 groups worked as partners
  - **All** partner groups got Gold stars
  - Only 8 out of 20 non-partner PS got Gold stars
- Your responses:
  - Learn more working alone: 7
  - Finish faster working alone: 5
  - Couldn't find anyone to work with: 2
  - Wanted to work with PS1/PS2 partner: 4
  - Started too late: 5

Quiz Responses

- How fast:
  - Way too fast: 1
  - Too fast: 11
  - Just about right: 11
  - Too slow: 1
- Exam 1:
  - Very confident: 1
  - Confident: 4
  - Concerned: 12
  - Worried: 5
  - Terrified: 3

AC's review session:
Wednesday at 7
My office hours:
Tuesday, 3:30-4:30
Thursday, 11:30am-12:30pm
others by email request

What should be Different

Maybe I've missed it, but beyond a cursory explanation of the meaning of the title of this course on the first day, I don't feel like I have an idea of where we're going with the material we're learning. What will I understand in a month/at the end of this course?

It would be good to do some review from the pre think I would digest more that way; A little slower; slow down a little

We'll try to do this in Wednesday's class

I wish lecture was a little slower, more chance to think. Sometimes questions get answered before I've had a chance to think about it. Other than that, its not too bad, I'm definitely challenged by the problem sets, but it's interesting and I like the concepts.

More puzzles and logic problems; More interactivity in lectures (stuff like the puzzle); Maybe more interactive things like have us work on a procedure (ex. One of the sorts) in class and then go over an answer (I know it sounds tedious) something that reinforces concepts were's working on

less programming! I know this is CS and all, and I find the ideas/problems with CS fascinating. I am horrible at programming; More conceptual description of the concepts as they are introduced. Shorter problem sets so that it is more feasible to do them without a partner; Maybe a slower pace with some review every few classes. The problem sets take a long time for me and I find myself losing time I would spend studying for other subjects so I can finish a problem set in time. Maybe they could be shorter?

It would be interesting to work with other languages as well in addition to Scheme

PS7 and PS8 will use different languages

Other Response

"I decided to skip this problem set, since the lowest grade gets dropped because of conflicts with other courses"

This is **not** the intent of (from the Syllabus):

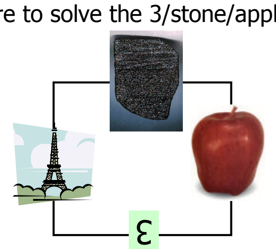
"For almost all students, doing the problem sets will b
G
a ys
u s
w l,
the weighting that is best for you is used."

The main point of the problem sets is to learn the material! (Note: student who answered this was the *only* one to also select "Way Too Fast" for question 2. If you don't do the PS, the course will definitely seem way too fast!

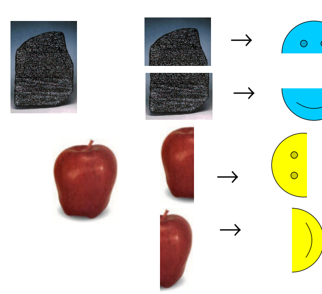
We will go over the other questions on the Quiz Wednesday.

Who cares about Smiley puzzles?

If we had a polynomial time procedure to solve the smiley puzzle, we would also have a fast procedure to solve the 3/stone/apple/tower puzzle:



3SAT → Smiley



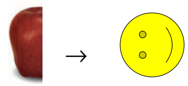
Step 1: Transform into smileys

Step 2: Solve (using our fast smiley puzzle solving procedure)

Step 3: Invert transform (back into 3SAT problem

Reductions

- Problem A *reduces* to Problem B if:
  - There is a polynomial time function f such that $A(x) = B(f(x))$
  - To reduce "3SAT" to "Smiley":
    - A = 3SAT
    - B = Smiley
    - f = tile replacements



- If A *reduces* B that means ***solving A is no harder than solving B*** since we can use a solution to B with f to solve A

The Real 3SAT Problem (also can be reduced to the Smileys Puzzle)

Propositional Grammar

Sentence ::= *Clause*

Sentence Rule: Evaluates to value of *Clause*

Clause ::= $\mathit{Clause}_1 \lor \mathit{Clause}_2$

Or Rule: Evaluates to **true** if **either** clause is **true**

Clause ::= $\mathit{Clause}_1 \land \mathit{Clause}_2$

And Rule: Evaluates to **true** iff **both** clauses are **true**

Propositional Grammar

Clause ::= ¬*Clause*

Not Rule: Evaluates to the opposite value of clause (¬**true** → **false**)

Clause ::= (*Clause*)

Group Rule: Evaluates to value of clause.

Clause ::= *Name*

Name Rule: Evaluates to value associated with *Name*.

Proposition Example

Sentence ::= *Clause*
Clause ::= $\mathit{Clause}_1 \lor \mathit{Clause}_2$ (or)
Clause ::= $\mathit{Clause}_1 \land \mathit{Clause}_2$ (and)
Clause ::= ¬*Clause* (not)
Clause ::= (*Clause*)
Clause ::= *Name*

$$a \lor (b \land c) \lor \neg b \land c$$

The Satisfiability Problem (SAT)

- Input: a sentence in propositional grammar
- Output: Either a mapping from names to values that satisfies the input sentence or **no way** (meaning there is no possible assignment that satisfies the input sentence)

SAT Example

Sentence ::= *Clause*
Clause ::= $Clause_1 \vee Clause_2$ (or)
Clause ::= $Clause_1 \wedge Clause_2$ (and)
Clause ::= $\neg Clause$ (not)
Clause ::= (*Clause*)
Clause ::= *Name*

SAT $(a \vee (b \wedge c) \vee \neg b \wedge c)$
→ { a: **true**, b: **false**, c: **true** }
→ { a: **true**, b: **true**, c: **false** }

SAT $(a \wedge \neg a)$
→ no way

The 3SAT Problem

- Input: a sentence in propositional grammar, where each clause is a disjunction of 3 names which may be negated.
- Output: Either a mapping from names to values that satisfies the input sentence or **no way** (meaning there is no possible assignment that satisfies the input sentence)

3SAT / SAT

Is 3SAT easier or harder than SAT?

It is definitely not harder than SAT, since all 3SAT problems are also SAT problems. Some SAT problems are not 3SAT problems.

Charge

- Wednesday’s class: recap in context of everything so far
- Friday: how Lorenz was really broken
- AC’s exam review is Wednesday, 7pm