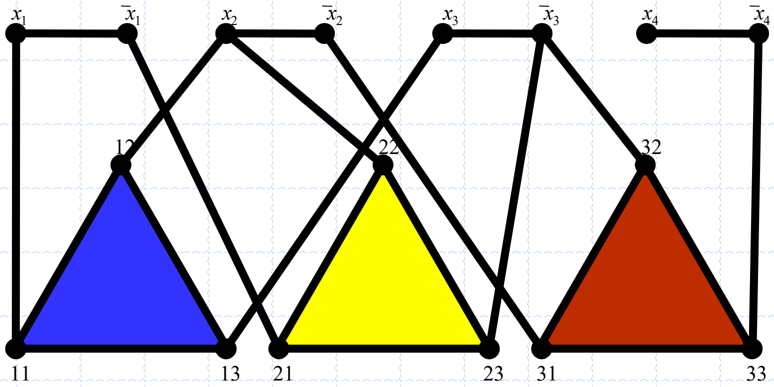
A Gentle Introduction to NP-Completeness



Michael T. Goodrich

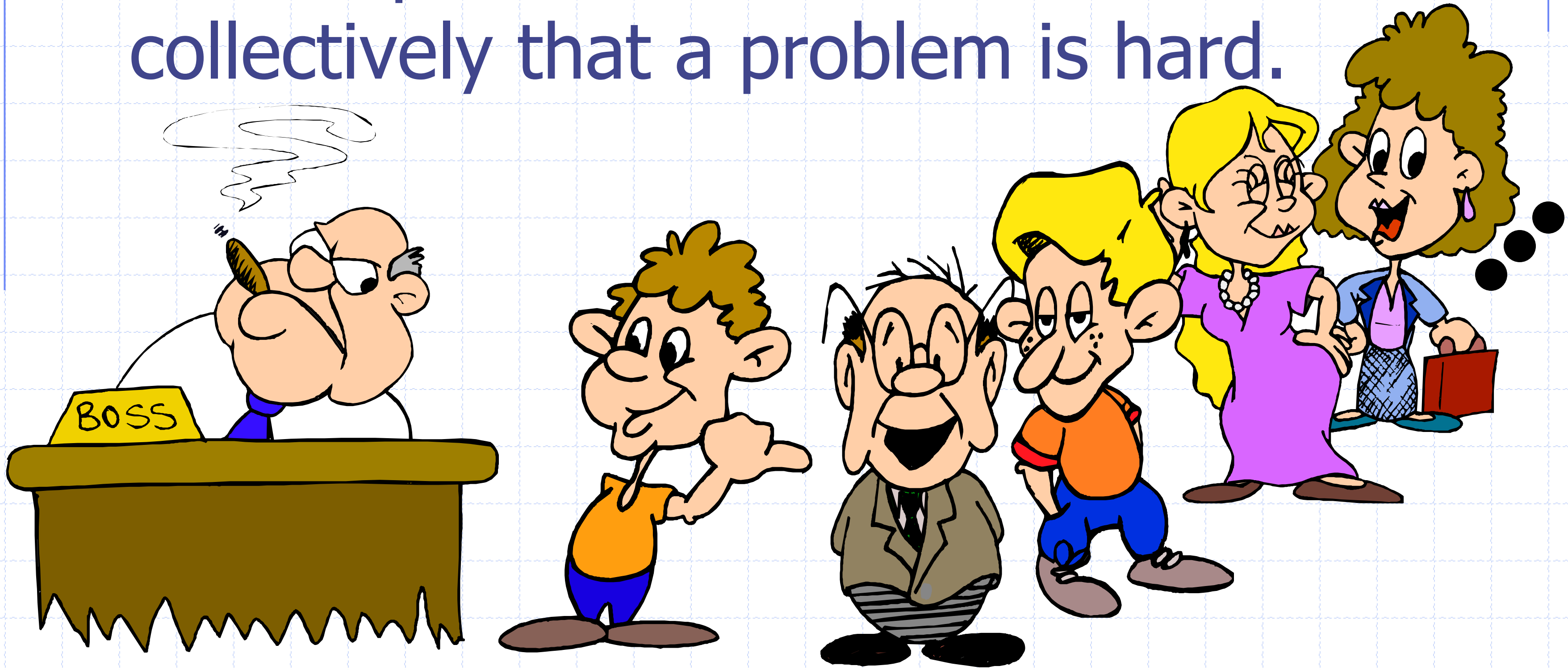
Dealing with Hard Problems

- What to do when we find a problem that looks hard…

I couldn’t find a polynomial-time algorithm;
I guess I’m too dumb.

Dealing with Hard Problems

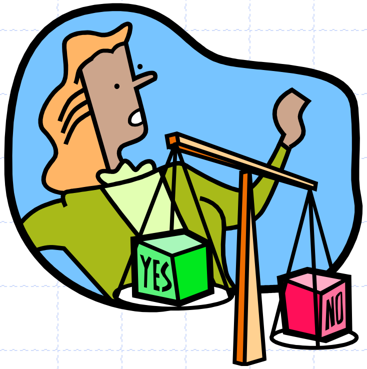
- NP-completeness let's us show collectively that a problem is hard.



I couldn't find a polynomial-time algorithm, but neither could all these other smart people.

(cartoon inspired by [Garey-Johnson, 79])

Polynomial-Time Decision Problems



- To simplify the notion of “hardness,” we will focus on the following:
  - Polynomial-time as the cut-off for efficiency
  - Decision problems: output is 1 or 0 (“yes” or “no”)
    - Examples:
    - Does a text T contain a pattern P?
    - Is the sequence, S, in sorted order?
    - Is it possible to graduate with a Computer Science major from UCI in 3 years without any AP credits?

Problems and Languages

- A **language** L is a set of strings defined over some alphabet Σ
- Every decision algorithm A defines a language L
  - L is the set consisting of every string x such that A outputs "yes" on input x.
  - We say "A **accepts** x'' in this case
    - Example:
    - If A determines whether or not a given graph G has an Euler tour, then the language L for A is all graphs with Euler tours.

The Complexity Class P

- A **complexity class** is a collection of languages
- P is the complexity class consisting of all languages that are accepted by **polynomial-time** algorithms
- For each language L in P there is a polynomial-time decision algorithm A for L.
  - If n=|x|, for x in L, then A runs in p(n) time on input x.
  - The function p(n) is some polynomial

The Complexity Class NP

- We say that an algorithm is non-deterministic if it uses the following operation:
  - Choose(b): chooses a bit b
  - Can be used to choose an entire string y (with |y| choices)
- We say that a non-deterministic algorithm A **accepts** a string x if there exists some sequence of choose operations that causes A to output "yes" on input x.
- NP is the complexity class consisting of all languages accepted by **polynomial-time non-deterministic** algorithms.

The Complexity Class NP Alternate Definition

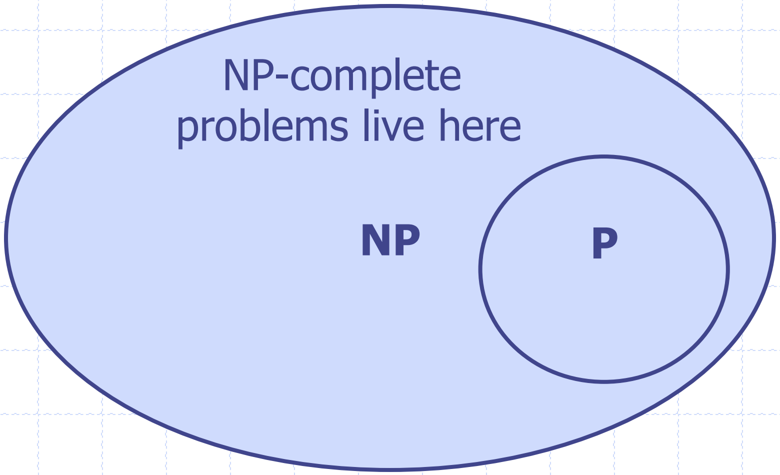
- We say that an algorithm B **verifies** the acceptance of a language L if and only if, for any x in L, there exists a certificate y such that B outputs "yes" on input (x,y).
- NP is the complexity class consisting of all languages verified by **polynomial-time** algorithms.

- We know: P is a subset of NP.
- Major open question: P=NP?
- Most researchers believe that P and NP are different.

NP-Completeness

- A language M is polynomial-time **reducible** to a language L if an instance x for M can be transformed in polynomial time to an instance x' for L such that x is in M if and only if x' is in L.
  - Denote this by $M \rightarrow L$.
- A problem (language) L is **NP-hard** if every problem in NP is polynomial-time reducible to L.
- A problem (language) is **NP-complete** if it is in NP and it is NP-hard.

Some Thoughts about P and NP



- Belief: P is a proper subset of NP.
- Implication: the NP-complete problems are the hardest in NP.
  - Why: Because if we could solve an NP-complete problem in polynomial time, we could solve every problem in NP in polynomial time.
- That is, if an NP-complete problem is solvable in polynomial time, then P=NP.
- Since so many people have attempted without success to find polynomial-time solutions to NP-complete problems, showing your problem is NP-complete is equivalent to showing that a lot of smart people have worked on your problem and found no polynomial-time algorithm.
- If you prove or disprove the P=NP, you will win $1 million.
  - See https://en.wikipedia.org/wiki/Millennium_Prize_Problems

Richard Karp

- Known for publishing a landmark paper proving 21 problems to be NP-complete.
  - One of these problems is related to the bin-packing problem we will study.
- 1985: Received the Turing Award.
- He was also the PhD advisor for UCI Professor Sandy Irani.

Image from https://en.wikipedia.org/wiki/Richard_M._Karp