P vs NP
How hard is a problem to solve?

Section 7.2
December 2, 2016

Polynomial vs Exponential
• Polynomial: $n^3$  Exponential: $3^n$
Complexity relationships between models

• Theorem 7.8: let $t(n) \geq n$, every $t(n)$ time multi-tape TM has an equivalent $O(t(n)^2)$ time single-tape TM.
  – polynomial difference

• Theorem 7.9: Every $t(n) \geq n$ time ND single tape TM has an equivalent $2^{O(t(n))}$ time deterministic single tape TM
  – exponential difference
The class P

• P is the class of languages

• Problems in class P

RELPRIME  Sipser, p 261

PROOF  The Euclidean algorithm $E$ is as follows.

$E = "\text{On input } (x, y), \text{ where } x \text{ and } y \text{ are natural numbers in binary:}"

1. Repeat until $y = 0$:
2. Assign $x \leftarrow x \mod y$.
3. Exchange $x$ and $y$.
4. Output $x$.”

Algorithm $R$ solves RELPRIME, using $E$ as a subroutine.

$R = "\text{On input } (x, y), \text{ where } x \text{ and } y \text{ are natural numbers in binary:}"

1. Run $E$ on $(x, y)$.
2. If the result is 1, accept. Otherwise, reject.”
Real Life

• Problems in class P are usually manageable on a real computer

\(-n^k\)

– though k=100 may introduce some practical problems
The class NP

- NP is the class of languages

Problems in class NP

Verifier

- A verifier of a language, A, is an algorithm, V, such that
  \[ A = \{ w \mid V \text{ accepts } <w, c> \text{ for some string } c \} \]
  where c is a certificate

  \[ |c| \text{ is polynomial in terms of } |w| \]
Clique, Sipser p 268

**Proof** The following is a verifier $V$ for CLIQUE.

$V =$ “On input $\langle G, k \rangle$, c;  
1. Test whether $c$ is a set of $k$ nodes in $G$  
2. Test whether $G$ contains all edges connecting nodes in $c$.  
3. If both pass, accept; otherwise, reject.”

**Alternative Proof** If you prefer to think of NP in terms of nondeterministic polynomial time Turing machines, you may prove this theorem by giving one that decides CLIQUE. Observe the similarity between the two proofs.

$N =$ “On input $\langle G, k \rangle$, where $G$ is a graph:  
1. Nondeterministically select a subset $c$ of $k$ nodes of $G$.  
2. Test whether $G$ contains all edges connecting nodes in $c$.  
3. If yes, accept; otherwise, reject.”

Subset-Sum Sipser p 269

**Proof** The following is a verifier $V$ for SUBSET-SUM.

$V =$ “On input $\langle S, t \rangle$, c;  
1. Test whether $c$ is a collection of numbers that sum to $t$.  
2. Test whether $S$ contains all the numbers in $c$.  
3. If both pass, accept; otherwise, reject.”

**Alternative Proof** We can also prove this theorem by giving a nondeterministic polynomial time Turing machine for SUBSET-SUM as follows.

$N =$ “On input $\langle S, t \rangle$:  
1. Nondeterministically select a subset $c$ of the numbers in $S$.  
2. Test whether $c$ is a collection of numbers that sum to $t$.  
3. If the test passes, accept; otherwise, reject.”
P vs NP

\[ P \subseteq NP \]

–unknown if the classes are unequal

• If \( P = NP \), then all problems in NP can be solved in polynomial time, if we are clever enough to find the right algorithm

NP-Complete

• NP-Completeness
  – set of problems in NP whose complexity is related to all problems in NP
  – if an NP-Complete problem can be shown to be in P, then \( P = NP \)
  – boolean satisfiability, for example
  – vertex-cover
  – clique
  – Hamilton Path
Recent Work

• Claim by Vinay Deolalikar (from HP Labs) that $N \neq NP$

• https://rjlipton.wordpress.com/2010/08/08/a-proof-that-p-is-not-equal-to-np/
  – Link to Deolalikar’s paper
  – Much discussion

• http://en.wikipedia.org/wiki/P-versus_NP_problem#Claimed_solutions

• https://rjlipton.wordpress.com/2010/08/12/fatal-flaws-in-deolalikars-proof/
  – Fatal flaws?