Algorithms vs. Lower bounds

- Algorithmic Theory
  - What we can compute
    - I can solve problem X with resources R
  - Proofs are almost always to give an algorithm that meets the resource bounds
- Lower bounds
  - How do we show that something can’t be done?

Theory of NP Completeness

The Universe

- P: Class of problems that can be solved in polynomial time
  - Corresponds with problems that can be solved efficiently in practice
  - Right class to work with “theoretically”
Decision Problems

- Theory developed in terms of yes/no problems
  - Independent set
    - Given a graph G and an integer K, does G have an independent set of size at least K
  - Network Flow
    - Given a graph G with edge capacities, a source vertex s, and sink vertex t, and an integer K, does the graph have flow function with value at least K

What is NP?

- Problems solvable in non-deterministic polynomial time . . .
- Problems where “yes” instances have polynomial time checkable certificates

Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula
Certificate: An assignment of truth values to the n boolean variables
Certifier: Check that each clause has at least one true literal,

\[
(\overline{x_1} \vee x_2 \vee x_3) \land (x_1 \vee \overline{x_2} \vee x_4) \land (x_1 \vee x_2 \vee x_3) \land (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4})
\]

Certificate \(1\)

\(x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1\)

Definition of P

Decision problems for which there is a polynomial time algorithm

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Algorithm</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTIPLE</td>
<td>Is (x) a multiple of (y)?</td>
<td>Grade school division</td>
<td>51, 17</td>
<td>31, 16</td>
</tr>
<tr>
<td>RELPRIME</td>
<td>Are (x) and (y) relatively prime?</td>
<td>Euclid’s algorithm</td>
<td>34, 39</td>
<td>34, 51</td>
</tr>
<tr>
<td>PRIMES</td>
<td>Is (x) prime?</td>
<td>Agrawal, Kayal, Saxena (2002)</td>
<td>53</td>
<td>51</td>
</tr>
<tr>
<td>EDIT-DISTANCE</td>
<td>Is the edit distance between s and t less than (K)?</td>
<td>Dynamic programming</td>
<td>neither</td>
<td>neper/tot</td>
</tr>
<tr>
<td>LSOLVE</td>
<td>Is there a vector (x) that satisfies (Ax = b)?</td>
<td>Gaussian elimination</td>
<td>(s)</td>
<td>(t)</td>
</tr>
</tbody>
</table>

Certificate examples

- Independent set of size \(K\)
  - The Independent Set
- Satisfiable formula
  - Truth assignment to the variables
- Hamiltonian Circuit Problem
  - A cycle including all of the vertices
- \(K\)-coloring a graph
  - Assignment of colors to the vertices

Certifiers and Certificates: Hamiltonian Cycle

HAM-CYCLE: Given an undirected graph \(G = (V, E)\), does there exist a simple cycle \(C\) that visits every node?

Certificate. A permutation of the n nodes.

Certifier. Check that the permutation contains each node in \(V\) exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.
Polynomial time reductions

• Y is Polynomial Time Reducible to X
  – Solve problem Y with a polynomial number of computation steps and a polynomial number of calls to a black box that solves X
  – Notations: $Y \leq_p X$

Lemmas

• Suppose $Y \leq_p X$. If $X$ can be solved in polynomial time, then $Y$ can be solved in polynomial time.

• Suppose $Y \leq_p X$. If $Y$ cannot be solved in polynomial time, then $X$ cannot be solved in polynomial time.

NP-Completeness

• A problem $X$ is NP-complete if
  – $X$ is in NP
  – For every $Y$ in NP, $Y \leq_p X$

• $X$ is a “hardest” problem in NP

• If $X$ is NP-Complete, $Z$ is in NP and $X \leq_p Z$
  – Then $Z$ is NP-Complete

Cook’s Theorem

• The Circuit Satisfiability Problem is NP-Complete

Circuit SAT

Find a satisfying assignment

Garey and Johnson

COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness
Michael R. Garey / David S. Johnson
History

• Jack Edmonds
  – Identified NP
• Steve Cook
  – Cook’s Theorem — NP-Completeness
• Dick Karp
  – Identified “standard” collection of NP-Complete Problems
• Leonid Levin
  – Independent discovery of NP-Completeness in USSR

P vs. NP Question

Populating the NP-Completeness Universe

• Circuit Sat $\leq_p$ 3-SAT
• 3-SAT $\leq_p$ Independent Set
• 3-SAT $\leq_p$ Vertex Cover
• Independent Set $\leq_p$ Clique
• 3-SAT $\leq_p$ Hamiltonian Circuit
• Hamiltonian Circuit $\leq_p$ Traveling Salesman
• 3-SAT $\leq_p$ Integer Linear Programming
• 3-SAT $\leq_p$ Graph Coloring
• 3-SAT $\leq_p$ Subset Sum
• Subset Sum $\leq_p$ Scheduling with Release times and deadlines

Sample Problems

• Independent Set
  – Graph $G = (V, E)$, a subset $S$ of the vertices is independent if there are no edges between vertices in $S$

• Vertex Cover
  – Graph $G = (V, E)$, a subset $S$ of the vertices is a vertex cover if every edge in $E$ has at least one endpoint in $S$

Cook’s Theorem

• The Circuit Satisfiability Problem is NP-Complete

• Circuit Satisfiability
  – Given a boolean circuit, determine if there is an assignment of boolean values to the input to make the output true
Proof of Cook’s Theorem

- Reduce an arbitrary problem Y in NP to X
- Let A be a non-deterministic polynomial time algorithm for Y
- Convert A to a circuit, so that Y is a Yes instance iff and only if the circuit is satisfiable