Peculiar Problems

1. Factoring: given an integer, find its prime factors.

Basis of RSA public key cryptography:
product of two large primes:
fast factoring would break the scheme

Testing primality is in P (recent result!)

Yes no question: does x have a factor between y and z? (Could factor by binary search)

Fastest algorithm: $2^{\tilde{O}(n^{1/3})}$ where n = # digits
(number field sieve)

→ Factoring in polynomial time by a quantum computer.

2. Linear Programming:

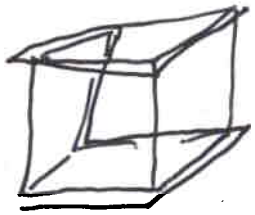
maximize cx subject to $Ax \leq b$

Classical algorithm: simplex (Dantzig, 1948)

exploits combinatorial structure,
iterative improvement to
local max = global max

Polytope can have exponentially many vertices.

Simplex fast in practice, standard methods exponential in the worst case (Klee-Minty squashed cube)



Is simplex fast non-deterministically?
(polytope diameter)

Best bound $\cong O(n^{\log n})$

(Weakly) polynomial algorithms: ellipsoid, interior point

Does LP have a strongly polynomial algorithm?

(polynomial in # arithmetic and logical ops, independent of number sizes)

Interesting special case: flows with gains:

maximize flow from s to t subject to capacity constraints and proportional losses:

if flow f enters edge e, flow $\alpha_e f$ leaves e.

3. Graph isomorphism:

Given two undirected graphs, is there a 1-1 mapping between their vertex sets that preserves adjacency?

Linear time for trees, planar graphs

Polynomial for fixed degree or fixed genus.

Intelligent backtracking using equal neighborhoods is fast in practice

Polynomial-time in general?

(Not known to be NP-complete.)

4. Boolean formula dualization (monotone case)

Given a monotone CNF formula and a monotone DNF formula, are they equivalent?

$$(a \vee b) \wedge (c \vee d) \equiv (a \wedge c) \vee (a \wedge d) \vee (b \wedge c) \vee (b \wedge d)$$

(De Morgan's laws)

Explicit application of De Morgan's law can blow up because of redundancy.

Best algorithm: $n^{O(\log n / \log\log n)}$ time