Lecture 31: Wednesday April 2, 2003

today

- NP completeness

announcements

- final exam

NP-Completeness [CLRS Ch 34]

- PTA: polynomial time algorithm
- so far, all algorithms we have seen are PTAs
- but . . . for many problems
  - no known PTA
  - best known alg'm exponential time
  - can sometimes show hardness results (namely, if PTA is found, it implies PTA for huge class of problems)

basic concepts

- (abstract) problem
- decision problem
- the classes P, NP, co-NP
- the satifiability problem (SAT)
- completeness: the class NP-c
- Cook's theorem: SAT $\in$ NP-c
- Karp's consequences: these 21 problems $\in$ NP-c
  - 3-CNF-SAT
  - graph 3-colouring
  - graph Hamiltonian
  - ...
- today: know thousands of problems $\in$ NP-c
- how to show a problem $\in$ NP-c

(abstract) problem

- two parts
  - a set of problem instances — the inputs
  - a query — the question
- instance solution: query answer for that instance — the output
- example of a problem: single-source shortest path problem
  - instance: all (G, s), with G a weighted graph and s a starting vertex
  - query: for the input weighted graph G and starting vertex s, for each vertex v, what is the length of a shortest $s - v$ path?
- decision problem: each solution is yes/no
- example of a decision problem: Hamiltonian graph problem
  - instances: all graphs
  - query: does the input graph G have a Hamiltonian cycle?

the class of problems P

- decision problem
- some alg'm solves problem in time poly'l in size of the problem
- problem size?
  - assume each instance represented as binary string
  - e.g. graph: string formed from adjacency matrix
  - size of instance is length of binary string
- Hamiltonian graph problem not known to be in P

the class of problems NP

- should be called PYV (polynomially yes-verifiable)
- decision problem
- for every instance with answer yes, there is a proof that the answer is yes which can be verified in polynomial time
- example: Hamiltonian graph problem is in NP
  - if G is Hamiltonian, it has a Ham'n cycle $(v_1, \ldots, v_n)$
  - it doesn't matter how long it takes to find the cycle
  - what matters: if you show the cycle to someone else ...
  - in $O(n^2)$ time, they can verify that it is a Ham'n cycle

- is the problem 'given a graph, is it not Hamiltonian?' in NP?

the class of problems co-NP

- should be called PNV (polynomially no-verifiable)
- decision problem
- for every instance with answer no there is a proof that the answer is no which can be verified in polynomial time
- example: "is this number prime?" is in co-NP
  - if instance (number x) is not prime, then $x = ab$ for some $a, b \geq 2$,
  - it doesn't matter how long it takes to find a, b
  - what matters: if you show a, b, x to someone else ...
  - in polynomial time, they can check that $a \times b = x$

P is in NP and co-NP

- if a problem is in P, it can be solved in polynomial time
- the solving algorithm can be used to verify any yes answer
- so the problem is in NP
- the solving algorithm can be used to verify any no answer
- so the problem is in co-NP

some well-known problems in NP

- k-clique: given a graph, does it have a k-clique?
- k-independent set: given a graph, does it have a k-ind. set?
- k-colouring: given a graph, can the vertices be coloured with k colours (so that adjacent vertices get different colours)?
- satisfiability (SAT): given a boolean expresssion, is there an assignment of truth values (T/F) to variables, so that the expression is satisfied (evaluates to T)?

polynomial time reduction

- a decision problem Π_1 is *polynomially reducible* to a decision problem Π_2 if there is a polynomial time transformation function t which maps instances of Π_1 to instances of Π_2 s.t. s.t. for all instances x of Π_1, the answer to x is the same as the answer to $t(x)$
- e.g. k-independent set is reducible to k-clique
  proof: X independent set in G iff X clique in $\overline{G}$
- e.g. 3-sat is reducible to sat — trivial exercise
- e.g. sat is reducible to 3-sat — non-trivial exercise