

# CS 573: Graduate Algorithms, Fall 2008

## Homework 1

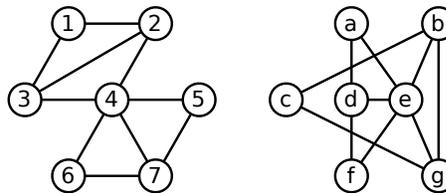
Due at 11:59:59pm, Wednesday, September 17, 2008

---

For this and all future homeworks, groups of up to three students may submit a single, common solution. Please neatly print (or typeset) the full name, NetID, and alias (if any) of every group member on the first page of your submission.

---

1. Two graphs are said to be *isomorphic* if one can be transformed into the other just by relabeling the vertices. For example, the graphs shown below are isomorphic; the left graph can be transformed into the right graph by the relabeling  $(1, 2, 3, 4, 5, 6, 7) \mapsto (c, g, b, e, a, f, d)$ .



Two isomorphic graphs.

Consider the following related decision problems:

- **GRAPHISOMORPHISM**: Given two graphs  $G$  and  $H$ , determine whether  $G$  and  $H$  are isomorphic.
- **EVENGRAPHISOMORPHISM**: Given two graphs  $G$  and  $H$ , such that every vertex in  $G$  and  $H$  has even degree, determine whether  $G$  and  $H$  are isomorphic.
- **SUBGRAPHISOMORPHISM**: Given two graphs  $G$  and  $H$ , determine whether  $G$  is isomorphic to a subgraph of  $H$ .

- Describe a polynomial-time reduction from **EVENGRAPHISOMORPHISM** to **GRAPHISOMORPHISM**.
- Describe a polynomial-time reduction from **GRAPHISOMORPHISM** to **EVENGRAPHISOMORPHISM**.
- Describe a polynomial-time reduction from **GRAPHISOMORPHISM** to **SUBGRAPHISOMORPHISM**.
- Prove that **SUBGRAPHISOMORPHISM** is NP-complete.
- What can you conclude about the NP-hardness of **GRAPHISOMORPHISM**? Justify your answer.

[Hint: These are all easy!]

- A *tonian path* in a graph  $G$  is a path that goes through at least half of the vertices of  $G$ . Show that determining whether a graph has a tonian path is NP-complete.
  - A *tonian cycle* in a graph  $G$  is a cycle that goes through at least half of the vertices of  $G$ . Show that determining whether a graph has a tonian cycle is NP-complete. [Hint: Use part (a).]
- The following variant of 3SAT is called either **EXACT3SAT** or **1IN3SAT**, depending on who you ask.

Given a boolean formula in conjunctive normal form with 3 literals per clause, is there an assignment that makes *exactly one* literal in each clause TRUE?

Prove that this problem is NP-complete.

4. Suppose you are given a magic black box that can solve the MAXCLIQUE problem *in polynomial time*. That is, given an arbitrary graph  $G$  as input, the magic black box computes the number of vertices in the largest complete subgraph of  $G$ . Describe and analyze a *polynomial-time* algorithm that computes, given an arbitrary graph  $G$ , a complete subgraph of  $G$  of maximum size, using this magic black box as a subroutine.
5. A boolean formula in *exclusive-or conjunctive normal form* (XCNF) is a conjunction (AND) of several *clauses*, each of which is the *exclusive-or* of several literals. The XCNF-SAT problem asks whether a given XCNF boolean formula is satisfiable. Either describe a polynomial-time algorithm for XCNF-SAT or prove that it is NP-complete.
- \*6. [*Extra credit*] Describe and analyze an algorithm to solve 3SAT in  $O(\phi^n \text{poly}(n))$  time, where  $\phi = (1 + \sqrt{5})/2 \approx 1.618034$ . [*Hint: Prove that in each recursive call, either you have just eliminated a pure literal, or the formula has a clause with at most two literals. What recurrence leads to this running time?*]

---

<sup>0</sup>In class, I asserted that Gaussian elimination was probably discovered by Gauss, in violation of Stigler's Law of Eponymy. In fact, a method very similar to Gaussian elimination appears in the Chinese treatise *Nine Chapters on the Mathematical Art*, believed to have been finalized before 100AD, although some material may predate emperor Qin Shi Huang's infamous 'burning of the books and burial of the scholars' in 213BC. The great Chinese mathematician Liu Hui, in his 3rd-century commentary on *Nine Chapters*, compares two variants of the method and counts the number of arithmetic operations used by each, with the explicit goal of find the more efficient method. This is arguably the earliest recorded *analysis* of any algorithm.