Title

Computer Science 604

Advanced Algorithms

Lecture 3: Review of NP-completeness and MORE, Cont'd

David Juedes

School of EECS

juedes@cs.ohiou.edu

NP-Completeness

The following problems (and many many more) are NP-complete.

SAT — Given a boolean formula Φ, determine whether Φ is satisfiable, i.e., whether Φ has a setting of its variables that causes it to evaluate to true.

3SAT — Given a boolean formula Φ in 3CNF (conjunctive normal form, 3 literals per clause), determine whether Φ is satisfiable, i.e., whether Φ has a setting of its variables that causes it to evaluate to true.

2SAT

Often, there is a sharp divide between “easy” and NP-complete. Consider the 2SAT problem:

2SAT — Given a boolean formula Φ in 2CNF (conjunctive normal form, 2 literals per clause), determine whether Φ is satisfiable, i.e., whether Φ has a setting of its variables that causes it to evaluate to true.

2SAT is solvable in polynomial-time via graph search techniques!

An aside on 2SAT

If we change our focus from decision to counting, then 2SAT becomes much harder.

#2SAT — Given a boolean formula Φ in 2CNF (conjunctive normal form, 2 literals per clause), determine the number of settings of the variables of Φ that cause Φ to be satisfiable.

This problem is #P-complete ("Sharp P-complete"). #P-completeness.

How does this relate to problem 209 from Project Euler?

NP-completeness, cont'd

The following problems are also NP-complete:

Independent Set — Given a graph $G = (V, E)$ and an integer k, determine whether G has an indepedent set of size k.

Vertex Cover —- Given a graph $G = (V, E)$ and an integer k, determine whether G has a *vertex cover* V' of size k. V' is a vertex cover of G if for every edge $\{u, v\} \in E$, either $u \in V'$ or $v \in V'$.

Clique — Given a graph $G = (V, E)$ and an integer k, determine whether there is a clique of size k in G, i.e., whether there is a set of vertices V' of size k such that $G[V']$ is a complete graph.

NP-completeness, cont'd

Dominating Set — Given a graph $G = (V, E)$ and an integer k, determine whether G has an dominating set of size k, i.e., whether G has a set D of size k such that for all $v \in V - D$, there exists an edge $\{u, v\}$ such that $u \in D$.

Subset Sum — Given a set of integers $S = \{s_1, \ldots, s_n\}$ and a target t, determine whether there is a subset $S' \subset S$ such that

$$\sum_{s \in S'} s = t.$$

Notice that Subset Sum is "easier" than the other problems since it can be solved in pseudo-polynomial time.

Pseudo Polynomial-Time

Given a decision problem Π, let I be an instance of the domain of Π. Then, we write $max(I)$ for the value of the largest integer in the instance I.

Then, we say that Π is computable in pseudo-polynomial time if there is an algorithm running in time $p(|I| + max(I))$ that decides Π on all instances I.

Solving SubSet Sum in Pseudo Polynomial-Time

We can solve Subset Sum in polynomial-time when the values of the integers are small via dynamic programming. Let's work this out in class.