CS 170 DIS 11

Released on 2018-04-11

1 Reduction Review

The dominating set of a graph $G = (V, E)$ is a subset D of V, such that every vertex not in D is a neighbor of at least one vertex in D.
Let the Minimal Dominating Set problem be the task of determining whether there is a dominating set of size $\leq k$.
Show that the Minimal Dominating Set problem is NP-Hard. You may assume for this question that all graphs are connected.

2 Randomization for Approximation

Often times, extremely simple randomized algorithms can achieve reasonably good appromxiation factors. For each of the following, determine a randomized algorithm that achieves the given approximation factor.

1. Consider Max 3-SAT (given a set of 3-clauses find the assignment that satisfies as many of the as possible). Come up with a simple randomized algorithm that will achieve an approximation factor of $\frac{7}{8}$ in expectation. That is, if the optimal solution satisfies k clauses, your algorithm should come up with an assignment that satisfies at least $\frac{7}{8} * k$ clauses in expectation. You may assume that every clause contains exactly 3 distinct variables in it.

2. What can this tell us about any instance of Max 3-SAT?

3 Fermat's Little Theorem as a Primality Test

Recall that Fermat's Little Theorem states the following:
"For a prime p and a coprime with p, $a^{p-1} \equiv 1 \pmod{p}$."
Assume for a general (not necessarily prime) p, we want to determine if p is prime. It may be tempting to try to use Fermat's Little Theorem as a test for primality. That is, pick some random a and check if $a^{p-1} \pmod{p}$. If this is equal to 1, return that p is prime, else return that it is composite. In this question we will investigate how effective this method actually is.

1. Suppose we wanted to test if 15 was prime. What is a choice of a that would trick us into thinking it is prime? What is a choice of a that would lead us to the correct answer? For choices of a that trick us into believing p is prime, we often say that p is "Fermat pseudoprime" to base a.

2. Suppose there exists a single a in $\pmod{p}$ such that $a^{p-1} \not\equiv 1 \pmod{p}$, where a is coprime with p. Show that p is not Fermat pseudoprime to least half the numbers in $\pmod{p}$. How might we use this to make our algorithm more effective?

3. Given the improvement from the previous question, why might our algorithm fail to still be a good primality test?