Approximation Algorithms, exercises week 2

October 18, 2013

Properties of $O(\cdot)$

a) Is $2^{2^n} \in O(2^n)$?

b) Does it hold for all functions $f, g : \mathbb{N} \to \mathbb{N}$ that $f(n) \in O(g(n)) \Leftrightarrow \log(f(n)) \in O(\log(g(n)))$?

c) Does it hold for all functions $f, g : \mathbb{N} \to \mathbb{N}$ that $f(n) \in O(g(n))$ or $g(n) \in O(f(n))$?

The 2-SAT problem

The difference between 2-SAT and 3-SAT is that 2-SAT instances have two literals per clause instead of three.

a) Does there exist an algorithm that solves the problem of whether there exists a satisfying assignment for a 2-SAT instance in polynomial time?

b) Does there exist an algorithm that, when given a 2-SAT instance, finds an assignment that satisfies a maximum number of clauses in polynomial time?

The relation between the questions P=NP? and PO=NPO?

does P=NP imply PO=NPO? What about the other direction?

Euler tour

An Euler tour in a graph $G = (V, E)$ is a sequence of vertices $(v_{i_1}, \ldots, v_{i_n})$ such that $i_1 = i_n$ and for every $e \in E$ there exists exaclty one $1 \leq j < n$ such that $e = (v_{i_j}, v_{i_{j+1}})$. In informal language, an Euler tour in a graph is a tour which ends where it starts and uses every edge exaclty once.

a) Does there exist an algorithm which on input (V, E) decides in polynomial time whether there exists an Euler tour in that graph?

b) Does there exist an algorithm which constructs such a tour in polynomial time if it exists?

non-approximability

Can you find a problem in NPO such that if that problem would have an approximation scheme with complexity polynomial in $|I|$ and $\log(\frac{1}{\epsilon})$, then P=NP?