1 ETH and fine-grained complexity

Recall that ETH asks whether certain NP-complete problems require, essentially, exponential time. One rationale we discussed in class was that if *some* problem in NP requires time $2^{n^{\epsilon}}$ for $\epsilon > 0$, then every NP-complete problem L requires time $2^{n^{\epsilon_L}}$ for some $\epsilon_L > 0$. In the next question you're asked to show a similar result.

Question 1. *We say that a function $T(n)$ is* quasipolynomial *if $T(n) = n^{\mathrm{polylog}(n)}$.*[1] *Show that if some problem in* NP *cannot be solved in quasipolynomial time, then all* NP*-complete problems cannot be solved in quasipolynomial time.*

Let's turn to ETH itself. The next question is a sanity check, making sure you understand the point of the sparsification lemma.

Question 2. *Recall that* ETH *implies that it's hard to solve* 3SAT *on formulas with $O(n)$ clauses (i.e., there are hard formulas that are sparse and have only $O(n)$ clauses). Why is this non-trivial? What is the maximal number of clauses that a 3-CNF formula can have, and more generally that a k-CNF formula can have?*

In class & tutorials we saw that ETH implies that certain problems other than 3SAT cannot be solved in time $2^{\epsilon \cdot n}$, for some $\epsilon > 0$. The next question asks you to prove such a result; the proof is similar to the ones you saw.

Question 3. *Show that if* ETH *is true, then there is $\epsilon > 0$ such that there is no algorithm solving a system of $O(n)$ quadratic equations modulo 2 over n variables in time $2^{\epsilon \cdot n}$.*

A dominating set in a graph $G = (V, E)$ is a set of vertices $S \subseteq V$ such that every vertex is either in S or a neighbor of a vertex in S. In the k-dominating-set problem we are given a graph with n vertices and need to decide if there is a dominating set of size k. Observe that the problem can be solved in time $n^{O(k)}$.[2]

Question 4. *Show that if* ETH *is true then for every $\epsilon > 0$ and large enough k (depending on ϵ) the problem k-dominating-set requires time $n^{\epsilon \cdot k}$, for some $\epsilon > 0$.*

Hint: Consider a variation on the reduction from 3SAT *to k-CLIQUE. Partition the variables into k sets of size n/k, create a vertex for each partial assignment, and create another vertex for each clause in the formula. The challenge is thinking of how to add edges to the graph to complete the reduction.*

In the problem k-set-cover we are given sets $S_1, ..., S_n$ and need to decide if there are k sets $S_{i_1}, ..., S_{i_k}$ whose union covers all elements, i.e. $\cup_{j \in [k]} S_{i_j} = \cup_{i \in [n]} S_i$. Observe that the problem can be solved in time $n^{O(k)}$.

Question 5. *Show that if* ETH *is true then k-set-cover requires time $n^{\epsilon \cdot k}$, for some $\epsilon > 0$.*

Hint: Reduce dominating set to set cover. The universe elements will be the vertices, and each set corresponds to a vertex and all its neighbors.

[1] Note that $n^{\mathrm{polylog}(n)} \gg \mathrm{poly}(n)$.

[2] Recall that when working in fine-grained complexity we fix some convenient machine model, say multitape machines or RAM machines.