ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Exercise Set I, Computational Complexity 2018

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 Prove the general Time Hierarchy Theorem. That is consider our simplified proof and understand what $f(n)$ and $g(n)$ need to satisfy so that $\mathbf{DTIME}(f(n)) \subsetneq \mathbf{DTIME}(g(n))$.

Recall that the universal Turing machine can simulate any other Turing machine by increasing the running time by a logarithmic factor.

Solution:

2 We focused in the lectures on decision problems: for a considered language we want to, given $x \in \{0,1\}^*$, decide whether $x \in L$. This is rather general as for polynomial-time computation, we can solve the search problem if we can solve the decision problem. To make this concrete, let

$$L = \{\text{all satisfiable Boolean formulas}\}.$$

Show that if there is a polynomial-time Turing machine for deciding L then there is a polynomial-time Turing machine for the search problem:

On input a Boolean formula φ on variables $x_1, \ldots, x_n$, find a truth-assignment that satisfies φ.

3 *(*, Problem 3.1)* Show that the following language is undecidable:

$$\{\alpha \in \{0,1\}^* : M_\alpha \text{ is a machine that runs in } 100n^2 + 200 \text{ time}\}.$$

Hint: reduce from the halting problem, i.e., show that if you can decide the above problem then you can decide the halting problem.

4 Determine whether the following language is decidable or undecidable:

$$\{\alpha \in \{0,1\}^* : M_\alpha(\alpha) \text{ halts and accepts in } 100|\alpha|^2 + 200 \text{ time}\}.$$

Motivate your answer.

5 Let $\mathbf{EXP} = \cup_{c>1}\mathbf{DTIME}(2^{n^c})$. Prove $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$. (Embarrassingly, we do not know of any stronger relation between $\mathbf{NP}$ and deterministic time classes.)

6 Vertex Cover is the language that contains all pairs $\langle G, k\rangle$ such that the graph G has a subset S of at most k vertices so that every edge $e = \{u, v\}$ of G has at least one endpoint in S, i.e., $S \cap e \neq \emptyset$. Show that Vertex Cover is $\mathbf{NP}$-complete.

7 *(half *, Problem 2.8 in book)* Let HALT be the language that contains those strings $\alpha, x \in \{0,1\}^*$ for which the Turing machine M_α represented by α halts on input x. Show that L is $\mathbf{NP}$-hard. Is it $\mathbf{NP}$-complete?