Computational Complexity, I

1. Determine which of the following relations hold: $3^n = 2^{O(n)}$, $\log_2(n) = o(n)$, $n^2 = O(n \log_2(n))$, $1/n = o(1)$, $n = o(2n)$.

2. Analyse your favourite algorithm for sorting a string of numbers of length n in terms of the number of required comparisons. Express the result using 'O' and/or 'o' notation. (*) Invent an algorithm that requires only $O(n \log n)$ comparisons.

3. GCD(m, n) is the greatest d that divides both m and n. Analyse the time and space complexity of the Euclidean algorithm to find GCD(m, n). (The numbers m and n have decimal representations.) Is it polynomial?

4. Show that any TM is *polynomially equivalent* to a TM with the tape alphabet $\{0, 1, \sqcup\}$.

5. Show that the size of a graph representation by an adjacency matrix is polynomial in its number of vertices. What about a representation by the lists of vertices and edges? (See Sipser, p. 237.)

6. A graph G is *connected* if any two vertices in G can be connected by a path. Show that testing connectivity of a graph is in P.

7. Show that P as a class of languages is closed under union, intersection, concatenation and complementation.

8. Show that graph isomorphism testing is in NP. (It is an open problem if it is in P.)

9. Problem No. 7.33 from Sipser.

10. (Huiswerk) Show that testing whether a graph has a cycle is in P.