6.045J/18.400J: Automata, Computability and Complexity Prof. Nancy Lynch

Recitation 9: Time Complexity, P, and NP

April 12, 2007 *Elena Grigorescu*

Readings: Sections 7.1, 7.2, 7.3

Problem 1: Let's review the following new terms and concepts.

1. Time complexity. The complexity classes $TIME(f(n))$.
2. Asymptotic, worst-case analysis
3. Polynomial vs exponential bounds
4. The class P: the class of languages where membership can be *decided* in polynomial time.
5. The class NP: the class of languages where membership can be *verified* in polynomial time.

Problem 2: Let's get some practice with asymptotic bounds. Roughly, you can think of these notations as follows (see Section 7.1 for precise definitions):

1. **Big-O:** $a(n) = O(f(n))$ means that $a(n)$ is less than or equal to a constant multiple of $f(n)$ for every n, once n is sufficiently large (i.e., an "upper bound").
2. **Big-Ω:** $c(n) = \Omega(f(n))$ means that $c(n)$ is greater than or equal to a constant multiple of $f(n)$ for every n, once n is sufficently large (i.e., a "lower bound").
3. Θ**:** $d(n) = \Theta(f(n))$ means that $d(n) = O(f(n))$ and $d(n) = \Omega(f(n))$.
4. **Small-o:** $b(n) = o(f(n))$ means that $b(n) = O(f(n))$ and $b(n) \neq \Omega(f(n))$.

Now, answer TRUE or FALSE for each of the following.

1. $n^2 = O(n^2 + n)$.
2. $2^n = 5^{O(n)}$.
3. $n^{1000000} = o(1.0000001^n)$.
4. For $c_1 < c_2$, $n^{c_1} = o(n^{c_2})$.

Problem 3: Prove that NP is closed under the star operation.

Problem 4:Sipser: Theorem 7.20 Prove that the two definitions of NP (the one involving the verifier and the one involving a NTM) are equivalent.

Problem 5:(NP) Let $MAXCUT = \{\langle G, k\rangle \mid G = (V, E)$ is an undirected graph and V can be partitioned into disjoint sets V_L and V_R such that the number of edges in E with one endpoint in V_L and the other in V_R is at least $k\}$. Prove that $MAXCUT$ is in NP.

Problem 6: Describe the error in the following fallacious proof that $P \neq NP$. Consider an algorithm for the problem $3COLOR = \{\langle G\rangle \mid$ G is a graph that can be colored "properly" with at most 3 colors$\}$: "On input a graph G, try all possible colorings of the nodes with 3 colors. If any of these colorings is proper, accept. Else, reject." Clearly, this algorithm requires exponential time. Thus $3COLOR$ has exponential time complexity. Therefore $3COLOR$ is not in P. Because $3COLOR$ is in NP, it must be true that $P \neq NP$. (*Aha, thats it !! where is my million-dollar prize ?*)[1]

[1] http://www.claymath.org/millennium/P_vs_NP/