UVa - cs302: Theory of Computation	Spring 2008

Problem Set 6 - Complexity

Due: Thursday, 24 April (2:02pm)

This problem set focuses on material on computational complexity from Classes 21-24 and Sipser's book Chapter 7. Answer all 7 questions. For full credit, answers must be concise, clear, and convincing, not just correct. Please **staple** your answer sheets before class.

Honor Policy (same as PS5). As with previous problem sets, you may discuss and work on the problems with anyone you want. After your discussions, you must destroy any notes from the meetings and write up your own solutions based on your own understanding. This policy also applies to notes from problem-solving sessions that pertain to questions on the problem sets. You may preserve notes from these meetings that pertain to other problems or general questions, but should not use notes from these meetings that include answers to specific questions from the problem set.

Problem 1: Asymptotic Notation. The way we advocated using asymptotic notation to describe algorithm complexity in Classes 21 and 22 is different from how Sipser uses it in Section 7.1. Describe (at least) two ways in which the book's use differs from the recommended use in class, and illustrate each with a specific example from Sipser's text.

Problem 2: Algorithm Analysis. What is the asymptotic run time of a Turing machine that can decide this language from Exam 2, $A = 0^n 1^m 0^{n/m}$ where $n \geq 0$ and $m \geq 1$ (that is, the input is n zeros, followed by m ones, followed by n divided by m zeros)? A reasonable big-O bound with a convincing argument is worth full credit. A correct Θ bound for the stated problem (that is, an argument that there is no asymptotically faster algorithm exists) is worth bonus points, but be aware that an incorrect Θ bound is worse than a correct big-O bound.

Problem 3: Cyclical Turing Machines. Describe a problem for which a cyclical Turing Machine (as defined in Problem 2b of Exam 2) can solve asymptotically faster than it can be solved on a regular Turing Machine. The problem does not need to be an interesting problem, but your answer should include a clear and convincing argument, and explain why the problem is in **TIME($t(n)$)** for the cyclical Turing Machine by **not** in the same time complexity class for the regular TM.

Problem 4: Multidimensional Turing Machines. Sipser's Theorem 7.8 shows that for every $t(n)$ time multitape Turing machine there is an equivalent single-tape

Turing machine that has time in $O(t^2(n))$. Consider the similar question for a $t(n)$ time multi-dimensional Turing machine (as defined in Class 15). A correct proof for any polynomial bound is worth full credit. If you can prove that your bound is the lowest possible bound is worth at least 50 bonus points.

Problem 5: Unary Subset Sum. (Sipser's 7.16) Let *UNARY_SSUM* be the subset sum problem in which all numbers are represented in unary.

a. Why does the NP-completeness proof for *SUBSET_SUM* fail to show *UNARY_SSUM* is NP-complete?

b. Show that $UNARY\_SSUM \in$ P.

Problem 6: Genome Assembly. In order to assemble a genome, it is necessary to combine snippets from many reads into a single sequence. The input is a set of n genome snippets, each of which is a string of up to k symbols. The output is the smallest single string that contains all of the input snippets as substrings. For example, if the input is {`ACCAGAATACC`, `TCCAGAATAA`, `TACCCGTGATCCA`}, the output should be `ACCAGAATACCCGTGATCCAGAATAA`:

```
ACCAGAATACC
              TCCAGAATAA
         TACCCGTGATCCA
```

a. Prove that the genome assembly problem is in NP.

b. Prove that the genome assembly problem is NP-Complete.

c. Explain how the human genome was sequenced even though it involves solving an NP-Complete problem for a large input size. The human genome is about 3 Billion base pairs. Readers at the time were able to read about 700 bases per read fragment, and sequencing the genome involved aligning about 30 million fragments.

Problem 7: Complexity Congress. Write a short (no more than one page) essay explaining the P = NP question in a way that (1) would be understandable to a typical congressperson, and (2) that makes it clear why it is a compelling and interesting problem. Especially good answers will also convince the prospective congressperson why it matters to them. Note that unlike some fifth graders, most congresspeople (with the possible exceptions of Representatives Vern Ehlers, Bill Foster, Rush Holt, Jerry McNerney, and John Olver) should not be expected to already understand what Turing Machines, algorithms, complexity classes, polynomials, and nondeterminism are, among other things.