CS3230: Design and Analysis of Algorithms (Fall 2014)

Tutorial Set #6

[For discussion during Week 8]

S-Problems are due (outside Prof. Leong's office): Friday, 3-Oct, before noon.

OUT: 27-Sep-2014 **Tutorials:** Tue & Wed, 7, 8 Oct 2014

IMPORTANT: Read "Remarks about Homework".

Submit solutions to S-Problem(s) by deadline given above.
Prepare your answers to all the D-Problems in every tutorial set.

When preparing to present your answers,
- Think of a CLEAR EXPLANATION
- Illustrate with a good worked example;
- Describe the main ideas,
- Can you sketch why the solution works;
- Give analysis of running time, if appropriate
- Can you think of other (perhaps simpler) solutions?

Helpful Hints Series: A problem well understood is half solved ☺.

Routine Practice Problems *-- do not turn these in -- but make sure you know how to do them.*

R1. COMPOSITE = Given a natural number k, determine if k is a composite natural number?

Show that COMPOSITE is in NP.

R2. Show that testing whether two graphs G and H are isomorphic is in NP.

R3. Show that testing whether a graph G is a subgraph of another graph H is in NP.

S-Problems: (To do and submit by due date given in page 1)
Solve this S-problem(s) and submit for grading.

IMPORTANT: Write your NAME, Matric No, Tutorial Group in your Answer Sheet.

S1. [Understanding P and polynomial time reductions]

Recall that decision problems can be viewed as subsets of $\{0,1\}^*$ (the set of all finite length binary strings).

Let A be a decision problem. Show that if $A \in P$, then for every decision problem B, we have: $A \leq_P B$ (unless B or complement of B is empty).

D-Problems: Solve these D-problems and prepare to discuss them in tutorial class. You may be called upon to present your solution *or your best attempt at a solution*. Your solution presentation does NOT need to be fully correct, given your best attempt. The TA will help clarify and correct any issues or errors.

D1. [Self reducibility]

FACTORIZE. Given an integer x, find its prime factorization.

FACTOR. Given two integers x and y, does x have a nontrivial factor less than y?

1) Show that: FACTOR $\equiv_P$ FACTORIZE.

2) Show that: FACTOR is in NP $\cap$ co-NP.

D2. [Prove that P ≠ EXP via a "Diagonalization" argument]

Take it granted that there exists an (infinite) listing of all polynomial time algorithms. Let P_k be the k-th algorithm in this listing. Take it granted that there exists a universal program U such that U(x ; k) = P_k(x), for every sting x and number natural number k. The running time of U on input (x ; k) is polynomial in |x| + k. Consider the following algorithm:

```
diag-p (k)

{  if (U(k ;  k) == true)
   return false
   else
   return true
}
```

Show that diag-p is an exponential time algorithm and the language (a.k.a decision problem) it decides is different from any language in P. From this conclude that P ≠ EXP.

D3. [Understanding P and NP]

0/1 KNAPSACK problem:

We are given a knapsack with maximum capacity W and a set S consisting of n items. Each item i has some weight w_i and benefit value b_i (all w_i, b_i and W are natural numbers which are given in binary encoding).

Problem: How to pack the knapsack to achieve maximum total value of packed items?

It can be shown that 0/1 KNAPSACK problem is NP-complete.

Professor Smart claimed that he could solve the 0/1 KNAPSACK problem in time O(W n) (we will also see subsequently in the course a 'dynamic programming' algorithm with same running time). Thus Professor Smart claimed that 0/1 KNAPSACK problem is in P and that he has shown P=NP. Could you find a flaw in his argument?

D4. [A language and its complement]

For a language L, let $\overline{L}$ be its complement language, that is

$\overline{L}$ = {x : x is a binary string and x $\notin$ L.}

Show that L $\leq_P \overline{L}$ if and only if $\overline{L} \leq_P$ L.

D5. [P, NP and co-NP]

1. Show that if NP ≠ co-NP, then P ≠ NP.
2. Show that L is NP-complete if and only if $\overline{L}$ is co-NP complete. A language A is co-NP complete if A is in co-NP and for every language B in co-NP, B $\leq_P$ A.