Algorithms and Complexity

Exercise session 9

Repetition

Different exercises from old exams

This is not a standard exam template but just a collection of mixed exercises.

1. PSPACE is the complexity class consisting of all languages for which there exists a *deterministic* Turing machine that recognizes the language in polynomial *space* (memory). EXPTIME consists of all languages for which there exists a *deterministic* Turing machine that recognizes the language of exponential time. Show that PSPACE$\subseteq$EXPTIME.

2. (6p) [Classical first exercise in the theory part]
   Are these statements true or false? For each sub-task provide a correct answer 1 point and proven correct answer 2 points;

   a) The problem of determining whether a n-digit number is prime number, is in the complexity class co-NP.
   b) There exists a constant $c > 1$ such that $n^3 \in O(c^{\log n})$.
   c) Binary tree is usually implemented by inserting two pointers in each entry (`left` and `right`). When you implement ternary trees (where each node has three children), you can not use less than three pointers in each entry.

3. In a large organization such as KTH there are many groups of people such as teachers at NADA, teachers at F, students of the Algorithms and Complexity course , members of Teknologkoren, etc. Each individual is included in at least one group, but one can be in many groups. Now the president wants to create a group of representatives who can quickly disseminate information to all individuals at KTH. He wants each group to be represented in this group (ie at least one member of each group will be in the representative group), but he wants the representative group to be as small as possible.

   This is an example of a general problem which, given a set of groups, finds the smallest group of representatives.

   a) (1p) Formulate the problem mathematically as a set problem and describe it at the same time as a decision problem.
   b) (5p) Show that the problem is NP-complete.

4. (6p) Input to the optimization problem Max k-Cut is a graph G. A solution is a cut of the vertices of G in k groups. The problem is to find a solution that maximizes the number of edges in the intersection between the groups, that is, between the vertices belonging to different groups. For $k = 2$ the problem is thus the same as Max Cut.

   Describe a probabilistic approximation algorithm for Max k-Cutand analyze the expected value of objective function (ie the number of edges that go between the vertices belonging to different groups). Try to develop an algorithm where the expected value is at least $1 - 1/k$ of all edges, which means that approximation factor is $k/(k - 1)$.

5. a) (6p) MAX 2$\wedge$SAT is an optimization problem which is defined as a decision problem as follows.

   Input: A positive integer K between 1 and n and n clauses where each clause consists of one or two literals combined by the operator $\wedge$. Example: $x_1 \wedge \overline{x}_3$, $x_2 \wedge x_3$.
   Problem: Is there a variable assignment that satisfies at least K clauses?

Show that the decision version of MAX $2\wedge$SAT problem is NP-complete. You might reduce to it the MAX 2SAT - the equivalent problem with the operator $\vee$ instead of $\wedge$ - which is NP-complete.

b) (5p) Construct a probabilistic approximation algorithm that approximates the optimization problem MAX $2\wedge$SAT within a factor of 4 (on average).