Algorithms and Complexity
2013
Mästarprov 2: Complexity

Mästarprov 2 should be solved individually in written form and presented orally. No collaboration is allowed.

Written solutions should be handed in latest on **Friday, April 19th 17.00,** to Johan, Musard or in the mailbox at the student reception at Lindstedtsvägen 3, 4th floor. Be sure to save a copy of your solutions. Mästarprov 2 is a mandatory and rated part of the course. The test consists of four tasks. The test is roughly graded as follows: Two task correctly solved give an E. Three tasks correctly solved give a C and all tasks correctly solved give an A. You can read more about the grading criteria and the final grade on the course web page. The report should be written either in English or Swedish.

1. Different types of Exact Cover

The problem EXACT COVER can be formulated like this:

Input: A set $M = \{x_1, x_2, ..., x_n\}$. A family $F = \{F_1, F_2, ..., F_m\}$ of subsets of M.

Goal: Is there a subset $\{F_{i_1}, F_{i_2}, ..., F_{i_p}\}$ of F such that every element of M belongs to exactly one of the F_{i_j}:s?

In this problem we contrast this standard formulation with a variant of the problem. We will call this variant FIXED SIZE EXACT COVER (FSEC):

Input: A set $M = \{x_1, x_2, ..., x_n\}$. A family $F = \{F_1, F_2, ..., F_m\}$ of subsets of M. An integer K.

Goal: Is there a subset $\{F_{i_1}, F_{i_2}, ..., F_{i_K}\}$ of F of size K such that every element of M belongs to exactly one of the F_{i_j}:s? (All the F_{i_j}:s are assumed to be distinct.)

We know that EXACT COVER is NP-Complete but we might believe that FSEC is simpler. But, in fact, FSEC is also NP-Complete. We can show this by reducing EXACT COVER to FSEC. One problem with doing this is that it seems impossible to tell what K should be. We cannot tell beforehand what size an EXACT COVER, if there is one, should have. But there is a trick we can use. Given an instance (M, F) of EXACT COVER we can define an instance (M', F', $K = n + 1$) of FSEC. M' will be $M \cup A$ and F' will be $F \cup G$ where A and G are "dummy" sets disjoint from M and F and such that the elements of G are subsets of A. If A and G are chosen in a clever way (M, F) will have an exact cover if and only if (M', F') has a $n + 1$ size cover. How? That is for you to find out. Do this:

1. Show that FSEC is in NP.
2. Find out what A and G should be and by doing that, define the reduction.

3. Prove that the reduction is correct.
4. Show that the reduction can be done in polynomial time.

Now that you have done this, you have shown that FIXED SIZE EXACT COVER is NP-Complete.

2. A comparison of programs

In this problem we will compare the running time of programs. We will say that P_1 is *dominated* by P_2 if, for all inputs x, $P_1(x)$ halts before $P_2(x)$. More precisely we say that P_1 is *dominated* by P_2 if, for all inputs x

$$\begin{cases} P_1(x) \text{ and } P_2(x) \text{ both halt and } P_1(x) \text{ halts in a smaller number of steps than } P_2(x) \\ \textit{or} \\ P_1(x) \text{ halts and } P_2(x) \text{ does not} \end{cases}$$

Show that the problem of, give two programs P_1 and P_2, deciding is P_1 is dominated by P_2, is *undecidable*.

3. Rectangle puzzle

In this problem we will study a very simple type of puzzle. Imagine that we have a rectangular grid of squares of size 1×1. In this grid we have a rectangle of size $a \times b$ where a and b are positive integers. The rectangle is oriented to fit the grid and therefore consists of ab squares. We also have n small rectangles (pieces of the puzzle) of sizes $h_1 \times b_1, h_2 \times b_2, ..., h_n \times b_n$. We want to place all these pieces into the big rectangle so that they are aligned with the grid and do not overlap. If $h_i \neq b_i$ there are two possible orientations of rectangle i and so on. Is it possible to fit in all the pieces, i.e., lay the puzzle. Note that the pieces does not have to cover all of the big rectangle. We can formulate this as the problem RECTANGLE PUZZLE:

Input: Positive integers a, b. A set of n pairs $\{(h_1, b_1), (h_2, b_2), ..., (h_n, b_n)\}$, all with positive integer values.

Goal: Given a rectangle with corners $(0, 0), (0, b), (a, 0), (a, b)$ in the coordinate plane, is it possible to place n small rectangles of sizes h_i, b_i into the rectangle so that the corners of the rectangles are placed in integer positions and do not overlap? (The small rectangles can be rotated $90°$.)

Decide if this problem, can be solved efficiently or not.

4. Finding vertex covers

In this problem we will use the technique of reducing construction problems to optimization problems.(Compare with examples from lecture notes and exercise problems.) What we will do is to assume that we have an algorithm F such that it, given a graph G, computes the size of a minimal vertex cover of G. Write an algorithm F' such that F' uses calls to F and finds an optimal vertex cover. If the time-complexity of F is $T(n)$ where n is the size of the input, then the time-complexity of G should be $O(p(n)T(n))$ where p is a polynomial of low degree.