Problem Set 1 – Combinatorial Optimization 2012 – EPFL

These will be discussed in the exercise session on Tuesday, September 25th. None of them are to be handed in.

Review of Linear Programming

1. For the primal and dual linear program

$$\max\{c^T x : Ax \leq b,\ x \geq 0\}, \quad \min\{b^T y : A^T y \geq c,\ y \geq 0\},$$

complete the following table by writing *possible* or *impossible*, with justification.

↓ primal dual →	finite optimum	unbounded	infeasible
finite optimum			
unbounded			
infeasible			

2. Given an "oracle" algorithm that can tell you a feasible solution $x \in \mathbb{R}^n$ of any system of linear inequalities $Mx \leq m$, give an algorithm that solves

$$\max\{c^T x : Ax \leq b,\ x \geq 0\}.$$

3. Give the dual of the following linear program:

$$\begin{array}{lll} \text{maximize} & x_1 + x_2 + x_3 & \\ \text{subject to} & x_1 + 2x_2 - 3x_3 & \leq 1, \\ & x_1 - x_2 + 2x_3 & \geq 2, \\ & -x_2 + x_3 & = 1, \\ & x_1 \geq 0,\ x_2 \leq 0. & \end{array}$$

4. Given a set of blue points and red points in $\mathbb{R}^2$, write down a linear program for finding a line that separates the blue points from the red points, if possible.

5. Write down an integral program for the Independent Set Problem: Given an undirected graph, find a maximum subset of vertices that have no edges between them. Give an example of a graph for which the relaxation has a different optimum value from the integer program.
Give the dual of the relaxation of the integer program, and an example for which this dual has a different optimum value from its integer version.

Basic Graph Algorithms

You can describe the algorithms informally, as long as they are mathematically precise. You do not have to worry about running time, but the algorithm should be reasonably efficient.

6. Find an algorithm that, given a directed graph and two disjoint vertex sets S, T, returns a dipath between a vertex of S and a vertex of T, if one exists.

7. Find an algorithm that, given an undirected graph, determines if the graph is bipartite. If it is, it should give a bipartition, and if it isn't, it should give proof of that.

8. Find an algorithm that, given an undirected connected graph, returns an Euler circuit or proof that none exists.
An *Euler circuit* is a circuit that uses each edge of the graph exactly once.

9. (Harder) Give a 2-approximation algorithm that finds a maximum acyclic subgraph in a digraph.
A digraph is *acyclic* if it has no *directed cycle* (a connected subgraph with the in-degree and out-degree of every vertex equal to 1). Note that this is not the same as the underlying graph being acyclic.