Exam Algorithms and Networks 2015/2016

This is the exam for part I of Algorithms and Networks.

You have two hours for the exam. You may give your answers in Dutch or in English. Write clearly. You may consult four sides of A4 with notes.

Results used in the course or exercises may be used without further proof, unless explicitly asked.

Switch of your mobile phone. Use of your mobile phone during the exam is stricktly forbidden.

Each of the five tasks: 1, 2, 3, 4, 5 counts for 2 points.

Some parts are harder than others: use your time well, and make sure you first finish the easier parts!

Important: Use separate sheets for questions 1, 2 and 3, and for questions 4 and 5. You may use more than one sheet for a collection. Failure to do so may result in having your work graded later.

Good luck!

Question 1: Some short questions (1 + 1 point)

1. **Explain the difference between a Hamiltonian Circuit and an Euler Tour.**
2. **Give an example of an input to the *Stable Roommates* problem without a stable matching.**

Question 2: Rounding with desired outcomes (2 points)

For a car company, you have to solve the following problem.

Given is an $n \times n$ matrix A. Each value $A[i, j]$ is a real number. Also given are two integer arrays $R[1 \cdots n]$ and $C[1 \cdots n]$. These are the *desired row sums* and *desired column sums*. The question to be answered is the following: is it possible to round each number $A[i, j]$ such that for each row, the sum of the rounded numbers on the row equals the desired row sum, and for each column, the sum of the rounded numbers on the column equals the desired column sum. I.e., we want to find an $n \times n$ matrix A' with integer values, such that

- for each i, $1 \leq i \leq n$ and each j, $1 \leq j \leq n$, $A'[i, j] \in \{\lfloor A[i, j] \rfloor, \lceil A[i, j] \rceil\}$.
- for each i, $1 \leq i \leq n$: $\sum_{j=1}^{n} A[i, j] = R[i]$.
- for each j, $1 \leq j \leq n$: $\sum_{i=1}^{n} A[i, j] = C[j]$.

Argue that this problem can be solved in polynomial time.

Hint: you can model this as a flow problem. Note the remark in the preamble of the exam about 'Results used in the course ...' Be clear in your answer, e.g., if you model the problem as a flow problem: what vertices and edges does your graph have, with what capacities, etc.

Question 3: Matching (2 points = 0.4+0.4+0.4+0.8)

(In this question, we prove a lemma that is used in a faster algorithm for maximum matching in bipartite graphs.)

Given two sets A and B, the *symmetric difference* $A \oplus B = (A - B) \cup (B - A)$, i.e., it consists of the elements that are in exactly one of the two sets.

A path P in an undirected graph is *augmenting* with respect to a matching M, if all of the following hold:

- it is simple, i.e., uses each vertex at most once;
- the first vertex on P is not an endpoint of an edge in M;
- the last vertex on P is not an endpoint of an edge in M;
- edges in P alternately belong to M and M';
- it has at least one edge.

A vertex is *isolated* if it has degree 0.

A collection of paths is *vertex disjoint*, if there is no vertex that belongs to more than one path in the collection.

Suppose we have a bipartite graph $G = (V, E)$. Suppose we have two matchings M and M'. Consider the graph $G' = (V, M \oplus M')$.

1. **Show that every vertex in G' has degree at most two.**
2. **Is the following statement true? Explain.** Each connected component in G' is a simple path, a simple cycle or an isolated vertex.
3. Let P be a simple path in G'. **Is the following statement true? Explain.** Edges in P belong alternately to M or M'.
4. **Prove** that if $|M| < |M'|$, then G' has at least $|M'| - |M|$ vertex-disjoint augmenting paths with respect to M.

Question 4: Pick-up And Delivery Problem (2 points)

In this question, you are to create an algorithm for a pick-up and delivery problem for your delivery van. In this problem, we are given a set of packages P, a set of locations L, and a distance function $d : L \times L \to \mathbb{R}_{\geq 0}$ giving distances from every location in L to every other location in L.

Each package $p \in P$ needs to be transported from its pick-up location $s(p) \in L$ to its delivery location $t(p) \in L$. The packages are small, so we can safely assume that your delivery van is big enough to contain all the packages. As such, it can pick-up and transport as much packages as required before delivering them. The delivery van is located at the depot $l_0 \in L$, where it should also return after delivering all packages.

Given an algorithm that runs in $O(|P|^2 2^{|P|})$ time and computes the shortest route for the delivery van which can be used to picks up and deliver all the packages. Briefly explain why your algorithm is correct and why it runs in the given amount of time.

Question 5: Toll Roads - Bicriteria Shortest Paths (2 points)

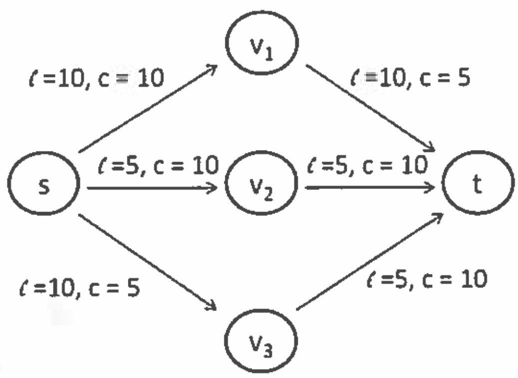
The European road network contains a lot of toll roads (roads on which you have to pay a fee to drive) especially for heavy trucks. Your client, a trucking company, is interested in computing shortest paths through the European road network. However, they are not only interested in the shortest paths as they certainly do not want to spend much money on toll.

You are given a directed graph $G = (V, A)$ together with a edge-length (road-length) function $\ell : A \to \mathbb{R}_{\geq 0}$ and a toll cost function $c : A \to \mathbb{R}_{\geq 0}$. For a given path P through G, the length of P is the sum of the lengths of all edges in P and the toll cost of P is the sum of the toll cost of the edges in P.

To save money, the trucking company is mainly interested in cheapest paths (least toll cost), and only if the toll cost of different paths are equal, then they are interested in the shortest paths. That is, they are interested in a path from a starting vertex $s \in V$ to a target vertex $t \in V$, that:

1. has least toll cost of all paths from s to t;
2. has the shortest length of all paths that satisfy the above.

In the example graph below, they are interested in the path through v_3. This because it is cheaper than the path through v_2 (even though the path through v_2 is shorter) and because it is shorter than the path through v_1.



Design an algorithm that, given G, ℓ, c, s and t, finds a most desired path from s to t in G. Analyse the running time of your algorithm.