Exam Algorithms and Networks 2013/2014

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.

$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$.

$\mathbf{Z}$ is the set of all integers.

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

Good luck!

1. Verifying a shortest paths algorithm

A company is making a program called PathPath for the single source shortest path problem, but the CEO suspects it to be buggy. Thus, the CEO wants a second program, that verifies the output of PathPath.

PathPath has as input a directed graph $G = (V, A)$ with for each arc $a \in A$ a length $\ell(a) \in \mathbf{Z}$, and a source $s \in V$. The lengths are integers, but are allowed to be negative; it is however given that G does not contain a cycle of negative length.

Now, the CEO wants an algorithm that

- has as input a directed graph $G = (V, A)$; a vertex $s \in V$; a length $\ell(e) \in \mathbf{Z}$ for each arc $a \in A$, and a value $d(v)$ for each $v \in V$.
- has as output "yes", if for all $v \in V$, $d(v)$ is the distance in G from s to v, and as output "no" otherwise.

(a) **Describe an efficient algorithm that solves this problem.** (You do not have to show correctness of your algorithm. Algorithms that are slower than necessary (w.r.t. O-notation) may get fewer points.)

(b) **How much time (in O-notation) does your algorithm take?**

2. TSP

Consider the travelling salesman problem with the following type of input. We have n cities, with for each city we have a pair of coordinates in the plane (x_i, y_i), $1 \leq i \leq n$. The distance between two cities is their Manhattan distance, i.e., $d((x_i, y_i), (x_j, y_j)) = |x_i - x_j| + |y_i - y_j|$.

(a) **Is there a polynomial time 2-approximation algorithm for TSP with this type of inputs?** I.e., is there a polynomial time algorithm that finds a tour whose distance is at most twice the optimal distance. **If so, give such an algorithm.** (If an algorithm discussed in the course can be used here, it is sufficient to give its name here. You do not need to prove the result.) **If not, explain why not.** (You may assume that $P \neq NP$.)

(b) Consider the algorithm that applies a 2-opt improvement as long as it is possible. **Show that this algorithm always produces a tour without crossings.**

3. Algorithm analysis: a variant of Schrijvers algorithm

Consider the following algorithm to find a perfect matching in a regular bipartite graph.

Given is a regular bipartite graph $G = (V, E)$, with all vertices of degree d. Initially, give each edge in E a weight of 1. Repeat the following steps until G does not contain a cycle such that all edges on the cycle have a weight that is at least 1.

- Find a cycle C in G such that each edge on C has weight at least 1.
- Partition the cycle C in two matchings, say M and N.
- Let α be the minimum weight of an edge on C.
- If M contains an edge with weight α, decrease the weight of all edges in M with α and increase the weight of all edges in N with α.
- Else (i.e., N contains an edge with weight α), decrease the weight of all edges in N with α and increase the weight of all edges in M with α.

(a) **Argue that an invariant of the algorithm is:**

$$\forall v \in V : \sum_{\{v,w\} \in E} \text{weight}(\{v, w\}) = d$$

(b) **Show that when the algorithm terminates, the edges with weight at least 1 form a perfect matching in G.**

(c) **Show that the algorithm uses $O(|E|)$ rounds.** (A 'round' is the sequence of steps, where we find one cycle, split this cycle in two matchings, and adjust the weights of edges on the cycle as described above.)

4. Modelling as a graph problem

Safe Facility Unlimited is a company, that guards buildings. At a number of locations in the city, the company has an office where a number of guards are usually sitting. Buildings are automatically guarded by electronic alarms. When an alarm rings, a car with two guards is sent to the building that caused the alarm.

The CEO of *Safe Facility Unlimited* wants to determine which office has to guard which building. Each office has a *capacity*, and an office guards at most its capacity number of buildings. Each office has a distance to a building, and offices can guard only buildings at a distance of at most five kilometres.

Given are the buildings, the offices, the distances, and the capacities.

The question that the CEO of SFU has is the following: does there exist (and if so, give it), an assignment of offices to buildings, such that

- when a building is assigned to an office, the building is not farther away than 5 kilometres.
- each office is assigned at most its capacity many buildings.
- each building is assigned to one office

(a) **Explain how this problem can be modelled as a graph problem.** (Explain clearly in words. You can clarify your description with help of an example if you want.)

(b) **Show that this problem can be solved in polynomial time.** (You may use here without further proof results and algorithms from the course.)

5. A hopping airplane with already sold tickets

An airplane company has an airplane that makes a number of hops. It starts at location s_0, and then goes to, in order: locations s_1, s_2, $\ldots$, s_{n-1} and s_n.

For each pair i, j, $0 \leq i < j \leq n$, there are $a_{i,j}$ passengers that want to go from s_i to s_j, and will pay $p_{i,j}$ euro for the flight in case the company transports them. For $b_{i,j}$ of these passengers, the company already sold a ticket. For the remaining $a_{i,j} - b_{i,j}$ passengers, the company can decide how many of these passengers it accepts to transport (from s_i to s_j.)

The plane can contain at most d passengers at each moment in time.

We assume that for all i, j, $0 \leq i < j \leq n$, $a_{i,j} \in \mathbf{N}$, $b_{i,j} \in \mathbf{N}$, $p_{i,j} \in \mathbf{N}$, and $a_{i,j} \geq b_{i,j}$. $d \in \mathbf{N}$.

(a) The company wants to decide how many passengers of each type to accept, such that all constraints are met, and the total profit is maximized. Consider the problem to decide if there is a choice that meets all the constraints, and if so, which choice optimizes total profit. **Explain clearly how this problem can be modeled** as a flow problem, a minimum cost flow problem or a polynomially solvable variant of these.

(b) **Illustrate your answer to (a) with a small example.**