Exam Algorithms and Networks 2016/2017

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

Each of the five questions: 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!

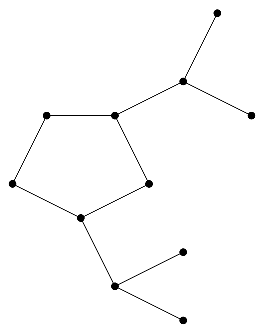
Important: Use separate sheets of paper for your answers to questions 1, 2 and 3, and for your answers to questions 4 and 5. You may use more than one sheet of paper for a collection. Failure to do so may result in having your work graded later.

Unless stated otherwise, n denotes the number of vertices of graph G, and m denotes the number of edges or arcs.

Good luck!

Question 1: Graph isomorphism and graph drawing (1 + 1 point)

a) How many automorphisms does the graph, given below have? Briefly explain your answer.



b) Explain in your own words a force directed method for drawing a graph. (Maximum 15 lines.)

Question 2: Maximal and Maximum Matchings (2 points)

a) Show that one can find a maximal matching in a graph in $O(n + m)$ time.

b) Is the following statement always true? (Write YES or NO.)

> Let $G = (V, E)$ be an undirected graph, and let $M \subseteq E$ be a maximal matching in G and let $M' \subseteq E$ be a maximum matching in G. Then $2 \cdot |M| \geq |M'|$.

c) Prove your answer to the previous question.

Question 3: Modelling as a Shortest Path Problem (2 points)

In a book-depot, one wants to put the books on shelves in the following manner. Each book has a height and width. The different heights are sorted in increasing order: $h_1, h_2, \ldots, h_n$, with $h_1 < h_2 < \cdots h_n$. The total width of all books of height h_i is l_i.

We now must order shelves to put all books on. In this case, the height of a shelf depends on the height of the books put on the shelf. There are shelves of height h_1, of height h_2, etc. We can put books on a shelf meant for books of a larger height, but we cannot put books on a shelf meant for books of a smaller height.

Suppose that ordering a length of x_i of shelves for books of height h_i costs $f_i + c_i \cdot x_i$; f_i are the fixed costs to buy this type of shelves; and c_i are the costs per unit of length for this type.

We know that $f_i \geq 0$ and $c_i \geq 0$. To save costs, one can possible order only for some heights shelves, and when not ordering shelves of a certain height, we put the books of that height on shelves of a larger height.

The question posed is: with what order can we place all books such that the costs are as small as possible. I.e., what kind of shelves do we order, and of what total-lengths.

Show how this problem can be modelled as a shortest paths problem.

Question 4: Forbidden Marriages (2 points)

A long time ago in a galaxy far far away, there was a small village with n young men and n young women ready to get married. In this hypothetical village, each man and each women has a preference list ranking all members of the other sex. Furthermore, there existed a village elder who had the authority to forbid certain marriages: pairs of a man and a woman who are *not allowed* to get married. In this question, we look for stable matchings in this setting.

We let M be the set of men, W be the set of women, and let $F \subseteq (M \times W)$ be the set of forbidden marriages: these forbidden marriages are pairs (m, w) that may not be part of a matching S. Because of the forbidden marriages, a stable matching in this setting does not have to include all the men and/or all the women. For example, if the village elder takes a fixed man m and sets $F = \{(m, w) : \forall w \in W\}$ then the man m must remain unmarried.

In this setting, we say that a matching S is *stable* if the following four types of instability do not exits:

1. There are two pairs (m_1, w_1) and (m_2, w_2) is S with $(m_1, w_2) \notin F$ such that m_1 prefers w_2 over w_1 and that w_2 prefers m_1 over m_2 (the usual type of instability).

2. There is a pair (m_1, w) in S and a man m_2 such that m_2 is not in a pair in the matching S, $(m_2, w) \notin F$ and w prefers m_2 to m_1.

3. There is a pair (m, w_1) in S and a woman w_2 such that w_2 is not in a pair in the matching S, $(m, w_2) \notin F$ and m prefers w_2 to w_1.

4. There is a man m and a woman w such that m and w both are not in a pair in the matching S while $(m, w) \notin F$ (two single people with nothing to prevent them from getting married).

Show how to modify the Gale-Shapley algorithm to find a stable matching in this setting. Argue why this modified algorithm is correct (you do not have to give a formal proof).

Question 5: An Inclusion/Exclusion Algorithm for List Colouring (2 points)

In this question, we consider the list colouring problem. This is a variant of the graph colouring problem, defined as follows. Let $C = \{1, 2, \ldots, k\}$ be a set of colours and let $G = (V, E)$ be an undirected graph with for every vertex $v \in V$ a set $L_v \subseteq C$ with allowed colours for that vertex. A *list colouring* of G is a colouring function $f : V \to C$ such that:

1. for all $v \in V$ the colour $f(v) \in L_v$ is on the list of colours for that vertex;
2. for all $\{u, w\} \in E$ we have that $f(u) \neq f(v)$ (it is a proper colouring).

In the *list colouring problem*, we are given G, the number of colours k and the lists L_v for every $v \in V$, and we are asked whether there exists a list colouring for G.

Theorem 1 *Let $z(G)$ be the number of sequences $(I_1, I_2, \ldots, I_k)$ where:*

1. *each I_i is an independent set in G;*
2. *each I_i only contains vertices v for which $i \in L_v$;*
3. *the independent sets I_i together cover all vertices in G ($\bigcup_{i=1}^{k} I_i = V$).*

Now, $z(G) > 0$, if and only if, G has a list colouring.

a) Prove the above theorem.

b) Give an algorithm that solves the list colouring problem in $\mathcal{O}^*(2^n)$ time and space.

   Hint: modify the inclusion/exclusion algorithm used for graph colouring, but be careful, you may have to do something multiple times.