Linköping University
Fall 2017
Communications and Transport Systems
Department of Science and Technology
Dr. Christiane Schmidt

Exam
Basic Logistic Algorithms
TNSL20
TEN1
24.10.2017

· Time: 14-18
· Number of questions: 8
· Total number of points: 100
· Grades: <50:UK, 50-67: 3, 67,5-82,5: 4, 83-100: 5
· Examinator: Christiane Schmidt
· Jourhavande lärare: Christiane Schmidt, tel 011-36 3212
· Hjälpmedel: Räknedosor som ej kan lagra text, alt. med tömda minnen är tillåtna. Ordböker engelsa-svenska är tillåtna. Inga andra hjälpmedel.
· Results will be published latest on November 7.

Please note:

· Carefully account for your computations and solution methods.
· Give reason/facts/motivation for all your claims.
· Always use the standard algorithms as presented in the course.
· You are allowed to use English-Swedish, Swedish-English dictionaries.
· You can write in either English or Swedish.
· Communications devices of any kind (phones, computers, etc.) are not allowed.
· This exam consists of 9 pages, plus 4 pages pseudocodes from the lectures.
· With 50 of 100 points you will pass the exam.
· You may not use a *red* pen for any written answers.
· You have 240 minutes to complete this exam.
· Sort your sheets of paper in the order of the given questions.
· Mark the problems you worked on on the envelope.
· Check how many papers you submit, and fill in the number on the envelope.

Problem 1: Shortest Paths **10 points**

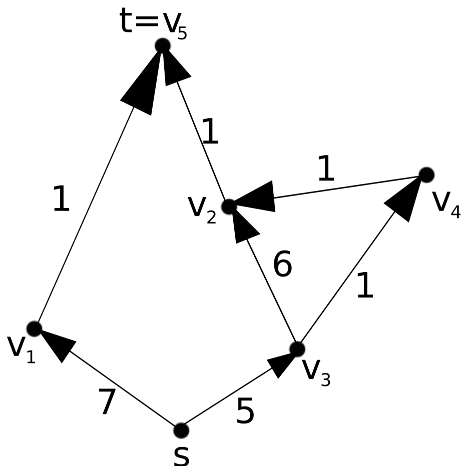

Figure 1: The graph G.

Use Dijkstra's algorithm to compute a shortest path from s to t in the graph G from Figure 1 For each iteration give the length and predecessors that *change*. If you can choose from more than one vertex, use the one with a smaller index.

Problem 2: **Maximum Flow** **5+5+5 points**

TransportSweden, a small company, tranpsorts—amongst others—palettes from s to t. Given the current remaining capacities of their trucks, they came up with a flow that allows them to transport 13 palettes from s to t. Now they are unsure whether that's as good as they can do, and ask you to help them:

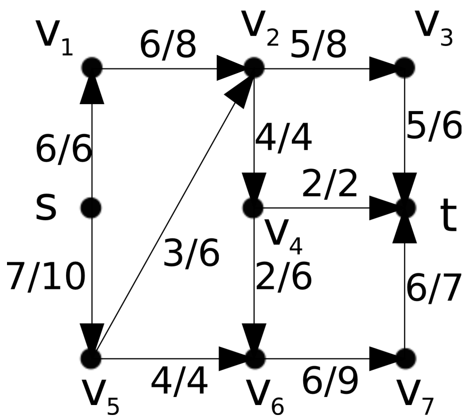

Figure 2: The network (G, c, s, t). The tupels at the edges have the form (flow/ capacity).

(a) Give the residual graph and the residual capacities for the network (G, c, s, t) from Figure 2.

(b) Execute an iteration of the algorithm of Edmonds and Karp. Give the augmenting path and the network with the new flow values.

(c) Is the flow you found optimal? Justify your claim.

Problem 3: Stable Marriage **10+5 points**

Consider the following preferences for Adam, Bert, Charles, David, Emilie, Frida, Greta, and Hannah:

Table 1: Men's preferences: leftmost is best, rightmost is least preferred.

Adam	Hannah	Emilie	Frida	Greta
Bert	Frida	Greta	Emilie	Hannah
Charles	Frida	Hannah	Greta	Emilie
David	Greta	Emilie	Hannah	Frida

Table 2: Women's preferences: leftmost is best, rightmost is least preferred.

Emilie	David	Adam	Charles	Bert
Frida	Adam	Charles	Bert	David
Greta	Adam	Bert	Charles	David
Hannah	David	Adam	Charles	Bert

(a) Use the appropriate algorithm to determine a stable matching for these eight people.

(b) Hannah suggests the following pairs: Adam + Emilie, Bert + Greta, Charles + Frida, Hannah + David. Is her suggestion a stable matching? Justify your claim.

Problem 4: Scheduling Conflicting Jobs **10 points**

At a small company 8 jobs need to be completed (j1, ... , j8), the company has four machines (M1, M2, M3, and M4) that are needed for these jobs, and four workers (Jane, Julia, Jack, and Jim) that are also needed for some of the jobs. The following table tells you exactly who and what is needed for which job. The execution of each single job takes exactly one working day.

job	machine	workers needed
j1	M1	Jane
j2	M1	Jack
j3	M2	Julia
j4	M3	Jack
j5	M1	Julia
j6	M2	Jim
j7	M4	Jane
j8	M4	Jack

Use the given information to construct the corresponding graph for scheduling conflicting jobs. Apply the appropriate algorithm from the lecture to tell the company after how many days they can complete all their jobs.

Problem 5: **Mail Delivery** **5+10 points**

Faster Mail plans a new route for its postman James. They are trying to work more efficiently, and ask you to help them replan the route. The streets James needs to cover are given by the graph G_1 in Figure 3. James must deliver mail along all these routes, that is, his route needs to pass each street at least once.

(a) Can James cover all of his streets without using any street twice? Give an argument for your statement.

(b) Use the appropriate algorithm to find the best rotue for James. What is the length of the resulting tour?

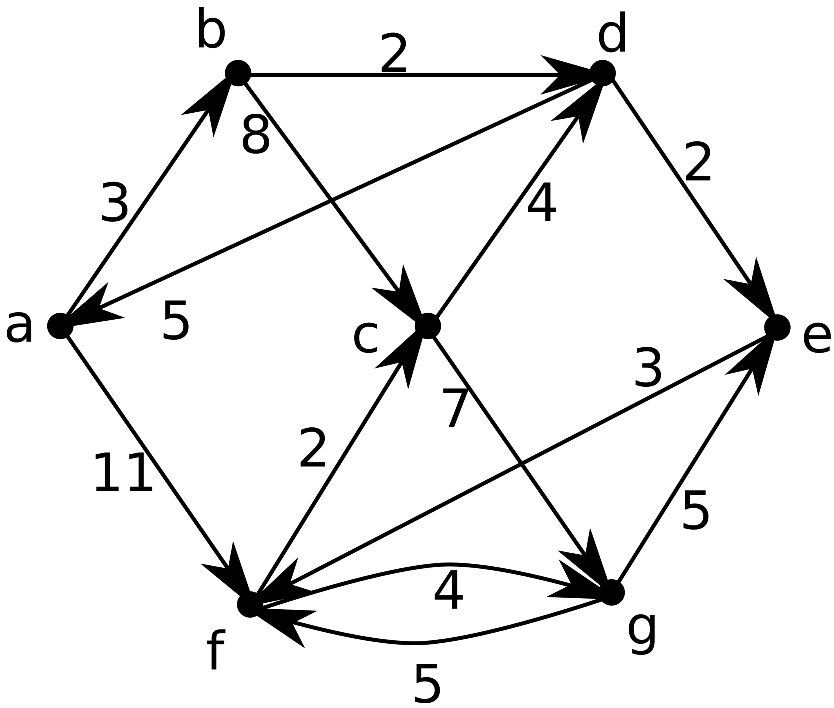

Figure 3: The graph G_1 representing the streets James needs to cover for his delivery route.

Problem 6: FOR loops **10 points**

Consider the following small algorithm:

$k = 0$
FOR $n = 1$ **TO** 10
    FOR $m = 1$ **TO** 50
        k=k+1

What is the value of k after running this algorithm?

Problem 7: **Maximum Matching in Bipartite Graphs** **7+8 points**

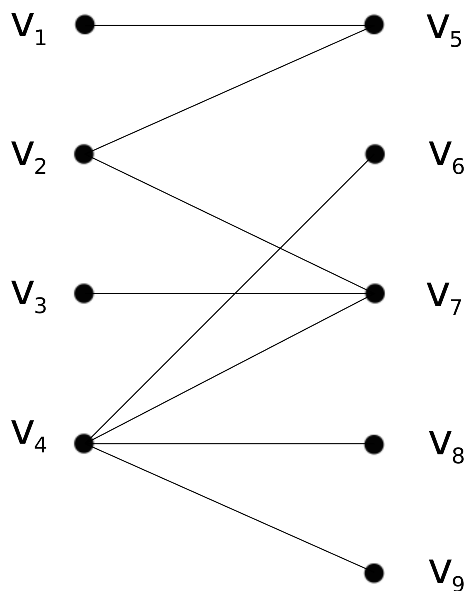

Figure 4: Graph G.

(a) Consider the bipartite graph G from Figure 4. Using the flow formulation we want to determine a maximum matching in G. Draw the network in which a maximum flow needs to be computed.

(b) Enter a flow with value 3 in the network from (a). (Hint: You do not need to use an algorithm to do so.) What does that tell you about a maximum matching?

Problem 8: **Maximum Bottleneck Paths** **10 points**

The company TruckLogistics often transports a wide load. Thus, it is mostly interested in the width of streets it will use to transport goods from a starting point to other locations.

We can model the street network as a graph $G = (V, E)$, where the edges are the streets, and vertices are intersections. The starting point is the vertex s, and we want to find the path P from s to any other vertex v in G, such that the narrowest street in P is as wide as possible. Thus, each edge becomes a weight that represents the width of the respective street. Of all possible paths that connects vertex s to vertex v, we want to find the path where the smallest weight of an edge (the narrowest street) is as large as possible. Assume for example, that we have two paths, P_1 and P_2, from s to v. The width of the edges along P_1 are $5, 7, 3, 9, 11$, and the width of the edges along P_2 are $6, 4, 4, 8, 7$. We only care about the narrowest streets, they are the bottleneck; this value is 3 for P_1 and 4 for P_2. Thus, of these two paths we would choose P_2 (wider vehicles will be able to take P_2), it has the larger bottleneck of the two.

Desing an algorithm for this problem by adapting an algorithm from the course, such that it computes such a maximum bottleneck path from a vertex s to all other vertices in the graph G.

Good Luck!!!