Final Exam Practice

Adam Piggott

April 2021

This document is not a practice exam. It contains some examples of exam-like problems, written in the style I prefer. Every topic we discussed in the course is examinable, but the problems here are all about weighted graphs. The purpose of this document is to give you a sense of the style in which problems may be written in your final. Together with the practice material for the midsemester exam and the midsemester exam itself, you have plenty of material to understand the style of problems I prefer to write.

A separate document has been uploaded to Wattle with practice problems in the style of the weekly assignment worksheets. Use that document to see pratice problems sampling from the entire course.

1. What is a weighted graph?

2. Describe three types of systems that may be modelled by weighted graphs.

3. In no more than 300 words, respond to the following:

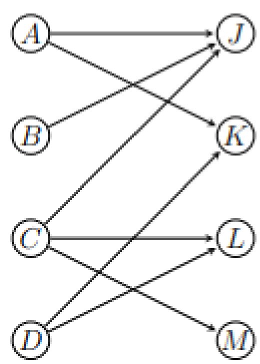
   > "Greedy algorithms never work."

   An excellent response will describe what greedy algorithms are, and use examples to illustrate whatever point(s) are made.

4. Complete the following table:

Name or description of algorithm discussed in the course	Input to algorithm	Output of algorithm
	Weighted connected graph G with n vertices.	Minimal Spanning Tree T for G and the Total Weight W of T
Dijkstra's Algorithm		
Nearest Neighbour Algorithm		
Vertex labelling algorithm for maximum flow		

5. Solve the following matching problem by adding a supersource s, supertarget t, and then applying the algorithm we learned in the course that finds the maximum flow through a transport network (pseudocode for this algorithm is given on the next page). The first incremental flow f_1 is shown in the first row of the table below. Write down the subsequent incremental flows in the table below (use only as many rows as necessary), and draw a digraph that represents the final matching.



incremental flow label	path of incremental flow	volume of incremental flow
f_1	$sAJt$	1

The vertex labelling algorithm for finding a maximum flow function for a transport network

Input: Transport network D with capacity function C.

Output: A maximum flow function $F_{\max}$ for the network.

Method: Initialise F to the zero flow F_0. Initialize i to 1.

For $i = 1, 2, \ldots$ carry out stage i below to attempt to build an incremental flow f_i.

If stage i succeeds, define $F_i = F_{i-1} + f_i$ and proceed to stage $i+1$.

If stage i fails, define $F_{\max} = F_{i-1}$ and stop.

Stage i:

1. If $i > 1$, mark up the amended edge flows for F_{i-1}.
2. Mark up the levels for F_{i-1}, as explained below.
3. If t is assigned a level, stage i will succeed, so continue.

   If not, then stage i fails, so return above to define $F_{\max}$ and terminate.
4. Mark up labels for F_{i-1} as follows until t is labelled:
   (a) Label each level 1 vertex v with sk_{v}, where $k_{\mathrm{v}} = S((s,v))$. (see below for definition of S)
   (b) If t has level 2 or more now work through the level 2 vertices in alphabetical order, labelling each vertex v with uk_u, where
      - u is the alphabetically earliest level 1 vertex with $(u,v) \in E(D)$ and $S((u,v)) > 0$,
      - k_{v} is the minimum of $S((u,v))$ and the value part of u's label.

   (c) If t has level 3 or more now work through the level 3 vertices in a similar manner and so on.
5. Let p_i be the path $\mathrm{u}_0\mathrm{u}_1 \ldots \mathrm{u}_n$ where $\mathrm{u}_n = \mathrm{t}$ and for $0 < j \leq n$
   u_j has label $\mathrm{u}_{j-1}k_j$.

   Define f_i to be the incremental flow on p_i with flow value k_n.

End of Method

Levels and labels: At each stage of the vertex labelling algorithm levels and labels are associated afresh with the vertices of the network.

The **level** of a vertex is determined iteratively as follows:

- The source vertex s always has level 0.
- If $e = (s, x)$ and $S(e) > 0$ then x has level 1.
- If x has level n and $S((x, y)) > 0$ then y has level $n + 1$ provided it has not already been assigned a lower level.

The **label** on a vertex v of level $n \geq 1$ has the form uk, where u is a vertex of level $n - 1$ and $(u, v) \in E(D)$ is an edge on the path for a potential incremental flow through v with flow value k.

The algorithm assigns labels in ascending order of levels, and in alphabetical order within levels.

The spare capacity function S
For vertices u,v of D, where D has capacity and flow functions C, F:

$$S((\mathrm{u,v})) = \begin{cases} C((\mathrm{u,v})) - F((\mathrm{u,v})) & \text{if } (\mathrm{u,v}) \in E(D) \\ F((\mathrm{v,u})) & \text{if } (\mathrm{v,u}) \in E(D) \\ 0 & \text{otherwise.} \end{cases}$$

When (v,u) $\in E(D)$, $S((\mathrm{u,v}))$ is called a **virtual capacity**.