DM582 Exercises - Sheet 3

Mads Anker Nielsen   Tobias Samsøe Sørensen

March 1, 2026

This document contains exercises from the course DM582 (spring 2025). Most exercises are from the book *Introduction to Algorithms, 4th edition* by Cormen, Leiserson, Rivest, and Stein (CLRS), the book *Algorithm Design, 1st edition* by J. Kleinberg and E. Tardos (KT), and the book *Discrete Mathematics and its Applications, 8th edition* by K. Rosen.

The solutions given here might differ from the solutions discussed in class. In class, we place more emphasis on the intuition leading to the correct answer. Please do not consider reading these solutions an alternative to attending the exercise classes.

References to CLRS refer to the book *Introduction to Algorithms, 4th edition* by Cormen, Leiserson, Rivest, and Stein.

References to KT refer to the book *Algorithm Design, 1st edition* by J. Kleinberg and E. Tardos.

References to Rosen refer to the book *Discrete Mathematics and its Applications, 8th edition* by K. Rosen.

References to BG refer to the book *Computer Algorithms: Introduction to Design and Analysis, 3rd edition* by Sara Baase and Allen Van Gelder.

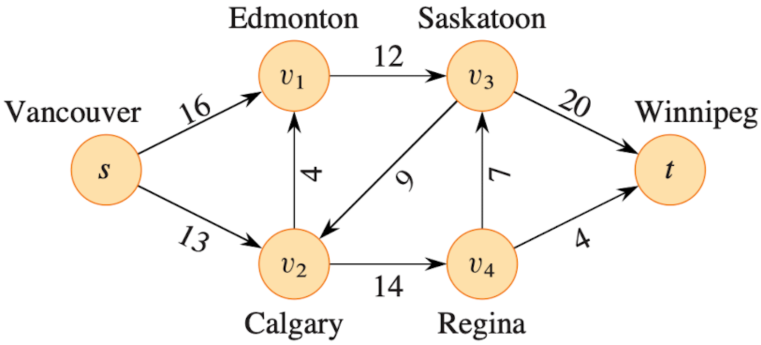
This document will inevitably contain mistakes. If you find some, please report them to your TA so that we can correct them.

Sheet 3

CLRS, 24.2-3

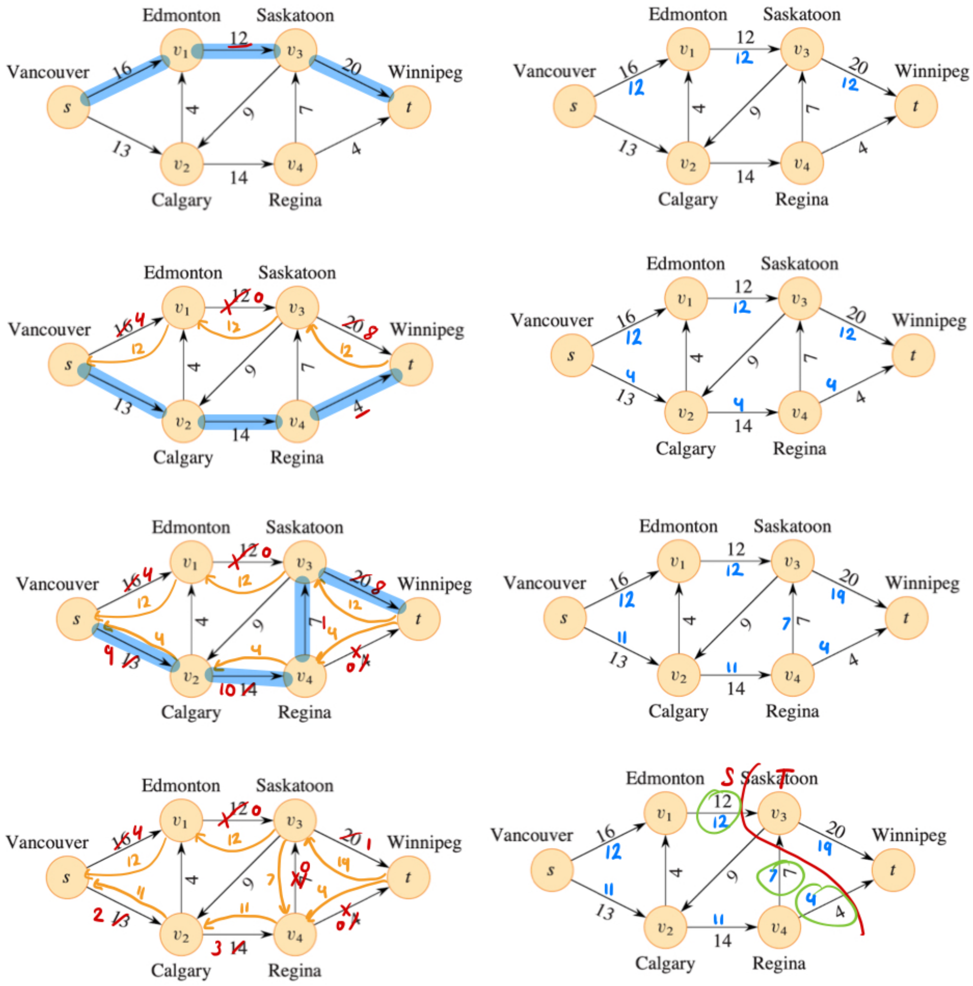
Exercise

Show the execution of the Edmonds-Karp algorithm on the flow network of Figure 24.1(a).



Suggested solution

See next page. I hope you can make out what it says.


Edmonton
Saskatoon
Vancouver
Winnipeg
Calgary
Regina
s
t
v_1
v_2
v_3
v_4
16
12
20
13
4
9
7
14
4
S
T

CLRS, 24.2-7

Exercise

Prove lemma 24.2.

Lemma 24.2. Let $G = (V, E)$ be a flow network, let f be a flow in G, and let p be an augmenting path in G_f. Define a function $f_p : V \times V \to \mathbb{R}$ by

$$f_p(u, v) = \begin{cases} c_f(p) & \text{if } (u, v) \text{ is on } p, \\ 0 & \text{otherwise} \end{cases}$$

Then, f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$.

Suggested solution

Note: Recall that an augmenting path is simple by definition.

First, we argue that f_p satisfies flow conservation. To see this, consider an arbitrary vertex $v \in V(G) \setminus \{s, t\}$. If v is not on p then the flow in and out of v is 0. If v is on p then, by the definition of f_p and since p is a simple path, there is exactly one arc uv entering v and exactly one arc vu' leaving v with $f_p(u, v) = f_p(v, u') = c_f(p)$. Hence, flow is conserved at v.

Next, we argue that f_p respects the capacity constraint c_f. An arc uv not on p has no flow and thus satisfies $f_p(u, v) \leq c_f(u, v)$ trivially. Let uv be an arbitrary arc on p. Then

$$\begin{aligned} f_p(u, v) &= c_f(p) \\ &= \min\{c_f(x, y) |\ xy \text{ on } p\} \\ &\leq c_f(u, v) \end{aligned}$$

where the last inequality holds since $c_f(u, v)$ is the residual capacity of a particular arc on p and $c_f(p)$ is the minimum residual capacity taken over all such arcs.

Lastly, we have $|f_p| = c_f(p) > 0$ since no arc in G_f has capacity 0 by definition and exactly one arc sv with non-zero flow leaves s and $f_p(s, v) = c_f(p)$.

CLRS, 24.2-8

Exercise

Suppose that we redefine the residual network to disallow edges into s. Argue that the procedure Ford-Fulkerson still correctly computes a maximum flow.

Suggested solution

The Ford-Fulkerson method computes the maximum flow by finding a sequence of augmenting (s, t)-paths in the residual network. No simple (s, t)-path can use an arc into s (since then it would not be a simple), so no implementation of the Ford-Fulkerson method is affected by the removal of such arcs.

CLRS, 24.2-9

Exercise

Suppose that both f and f' are flows in a flow network. Does the augmented flow $f \uparrow f'$ satisfy the flow conservation property? Does it satisfy the capacity constraint?

Suggested solution

$f \uparrow f'$ does satisfy flow conservation, but the capacity constraints are not at all respected. The definition of $(f \uparrow f')(u, v)$ (24.4) reduces to $(f \uparrow f')(u, v) = f(u, v) + f'(u, v)$ for all arcs $uv \in E(G)$ since we assume no antiparallel arcs. Thus, for any $v \in V(G)$

$$\begin{aligned}
\sum_{uv \in E(G)} (f \uparrow f')(u, v) &= \sum_{uv \in E(G)} f(u, v) + f'(u, v) \\
&= \sum_{uv \in E(G)} f(u, v) + \sum_{uv \in E(G)} f'(u, v) \\
&= \sum_{vu \in E(G)} f(v, u) + \sum_{vu \in E(G)} f'(v, u) \\
&= \sum_{vu \in E(G)} f(v, u) + f'(v, u) \\
&= \sum_{vu \in E(G)} (f \uparrow f')(v, u).
\end{aligned}$$

We can construct an example with only one arc st showing that the capacity constraints are not respected (we did this in class).

CLRS, 24.2-10

Exercise

Show how to find a maximum flow in a flow network by a sequence of at most $|E|$ augmenting paths. (Hint: Determine the paths after finding the maximum flow.)

Suggested solution

Let f be a maximum flow in G. Take any simple path P from s to t in G using only arcs with positive flow. If no such path exists, then $|f| = 0$ and thus the empty sequence of augmenting paths results in a maximum flow.

Otherwise, let α be the minimum flow on any arc on P and let f' be the flow obtained by subtracting α from the flow on each arc of P. Then f' is a flow in G with $|f'| = |f| - \alpha$. Furthermore, at least one arc (the one with flow α) has no flow in f' and thus will never be used again.

Repeating this process at most $|E|$ we obtain the desired paths.

CLRS, 24.2-11

Exercise

The edge connectivity of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how to determine the edge connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V + E)$ vertices and $O(E)$ edges.

Suggested solution

We can solve this problem as follows. Obtain G' by replacing every edge $\{u, v\}$ in G by a pair of antiparallel edges (u, v) and (v, u) and set the capacity of all edges to 1. We should then convert the network to an equivalent one with no antiparallel edges as done in CLRS Figure 24.2, though it is conceptually easier to think of the network as having antiparallel edges. Now, pick any $s \in V(G)$ as the source and, for all $t \in V(G) \setminus \{s\}$, compute the value $|f_{st}|$ of a maximum (s,t)-flow in G'. Let $f_m = \min_{t \in V(G) \setminus \{s\}} |f_{st}|$ be the lowest value of a maximum (s,t)-flow computed. We claim that f_m is the edge connectivity of G.

Let $F \subseteq E(G)$ be a minimum set of edges such that $G - F$ (G with the edges in F removed) is disconnected. Then there is some vertex t not reachable from s in $G - F$. Thus, F contains the edges of some (s,t)-cut (S, T) in G. Recalling that every edge in G is replaced by two antiparallel arcs in G' and that the capacity of all arcs is 1, we see that the capacity of (S, T) in G' is the number of edges in the cut (S, T) in G. Thus, by the max-flow min-cut theorem, we have $|f_{st}| \leq |F|$, Since $f_m = \min_{x \in V(G) \setminus \{s\}} |f_{sx}|$, we have $f_m \leq |f_{st}| = |F|$.

Lastly, observe that we cannot have $f_m < |F|$ since then, by the max-flow min-cut theorem, there would be a cut (S, T) in G' with capacity less than $|F|$, and the set of edges across (S, T) would disconnect G with fewer than $|F|$ edges, contradicting the minimality of F.

We conclude that $f_m = |F|$, which is what we wanted. Note that the conversion from Figure 24.2 at most doubles the number of edges, and creates at most one new vertex per edge, which guarantees an upper bound of $O(V + E)$ vertices and $O(E)$ edges in each of the $|V| - 1$ networks.

CLRS, 24.2-12

Exercise

You are given a flow network G, where G contains edges entering the source s. Let f be a flow in G with $|f| \geq 0$ in which one of the edges (v, s) entering the source has $f(v, s) = 1$. Prove that there must exist another flow f' with $f'(v, s) = 0$ such that $|f| = |f'|$. Give an $O(E)$-time algorithm to compute f', given f and assuming that all edge capacities are integers.

Suggested solution

We assume that f is an integer flow.

Find a simple 'path' p from s back to s in the residual network starting with the edge sv and augment the flow along this path by $1 \leq c_f(p)$ (which holds since f is an integer flow). s loses 1 unit of flow out and one unit of flow in, so the flow value remains the same.

CLRS, 24.2-13

Exercise

Suppose that you wish to find, among all minimum cuts in a flow network G with integer capacities, one that contains the smallest number of edges. Show how to modify the capacities of G to create a new flow network G' in which any minimum cut in G' is a minimum cut with the smallest number of edges in G.

Suggested solution

We obtain G' by adding a sufficiently small quantity $\epsilon > 0$ to the capacity of each arc (we will see how small). Let C be the capacity of a minimum cut and let m be the number of edges in a minimum cut with the fewest number of edges. If a minimum cut (S, T) contains $M > m$ edges, then the capacity of (S, T) in G' is $C + M\epsilon > C + m\epsilon$ and thus (S, T) is not a minimum cut in G'. Note that this holds for any $\epsilon > 0$.

If we pick ϵ sufficiently small, no cut that was previously not a minimum cut will become a minimum cut. Indeed, since all capacities are integers, a minimum cut in G has capacity at least 1 less than any non-minimum cut. Thus, if we pick ϵ such that the capacity of a minimum cut in G increases by less than 1, we obtain the desired property.

We show that $\epsilon < \frac{1}{|E|}$ is sufficiently small (in practice, we could pick $\epsilon = \frac{1}{|E|+1}$). Let (S, T) be a minimum cut in G with capacity C and m edges. Let C' be the capacity of (S, T) in G'. Then

$$C' = C + m\epsilon \leq C + |E|\epsilon < C + 1$$

since no cut contains more than $|E|$ edges.