MTAT.03.286: Advanced Methods in Algorithms **University of Tartu**

Final exam

Instructor: Dr. Vitaly Skachek January 16th, 2014

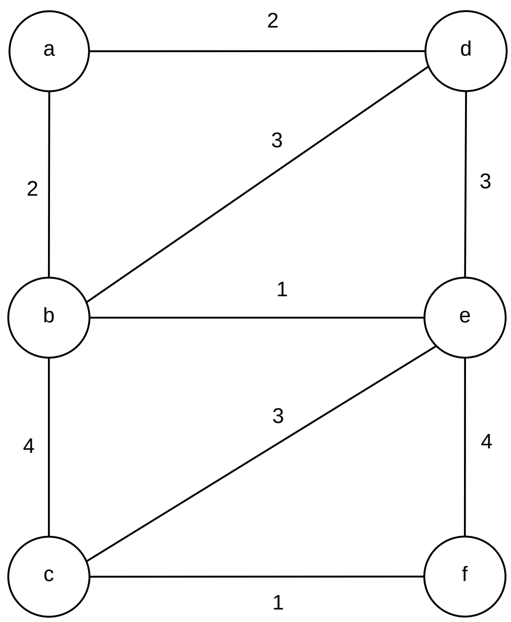
Student name: ____

Student ID: ____

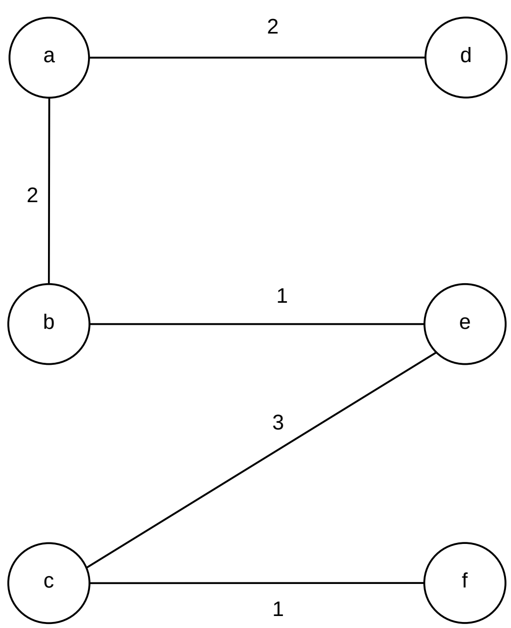
1. This exam contains 10 pages. Check that no pages are missing.
2. It is possible to collect up to 120 points. Try to collect as many points as possible.
3. Justify and prove all your answers (where applicable).
4. All facts and results that were proved or stated in the class can be used in your solution without a proof. Such results need to be rigorously formulated.
5. Any printed and written material is allowed in the class. No electronic devices are allowed.
6. Exam duration is 2 hours.
7. Good luck!

Question 1	
Question 2	
Question 3	
Question 4	
Total	

Question 1 (20 points). Find a minimum spanning tree in the following graph by using one of the algorithms studied in the class. Show all the stages of the algorithm run.



Answer The minimum spanning tree is as follows.



Its weight is 9.

Question 2 (30 points). You are given the following linear-programming (LP) problem:

$$\begin{array}{llcl} \textbf{max} & x_1 + x_2 + 2x_3 & & \\ \textbf{s.t.} & x_1 + x_2 + x_3 & \leq & 4 \\ & \quad\;\; 3x_2 + x_3 & \leq & 5 \\ & 2x_1 \quad\quad - x_3 & \geq & 0 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0 & & \end{array}$$

(a) Solve this LP problem by using the simplex method.

(b) Formulate the dual LP problem.

(c) *This part is cancelled.*

Answer (a) Rewrite the third inequality as $-2x_1 + x_3 \leq 0$. Then,

$$\left(\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 4 \\ 0 & 3 & 1 & 0 & 1 & 0 & 5 \\ -2 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline \textcircled{1} & 1 & 2 & 0 & 0 & 0 & 0 \end{array}\right) \begin{array}{l} \\ \\ (3) \leftarrow (3) + 2\cdot(1) \\ (\star) \leftarrow (\star) - (1) \end{array}$$

$$\Longrightarrow \left(\begin{array}{ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 4 \\ 0 & 3 & 1 & 0 & 1 & 0 & 5 \\ 0 & 2 & 3 & 2 & 0 & 1 & 8 \\ \hline 0 & 0 & \textcircled{1} & -1 & 0 & 0 & -4 \end{array}\right) \begin{array}{l} (1) \leftarrow (1) - \frac{1}{3}\cdot(3) \\ (2) \leftarrow (2) - \frac{1}{3}\cdot(3) \\ (3) \leftarrow \frac{1}{3}\cdot(3) \\ (\star) \leftarrow (\star) - \frac{1}{3}\cdot(3) \end{array}$$

$$\Longrightarrow \left(\begin{array}{ccc|ccc|c} 1 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & -\frac{1}{3} & \frac{4}{3} \\ 0 & \frac{7}{3} & 0 & -\frac{2}{3} & 1 & -\frac{1}{3} & \frac{7}{3} \\ 0 & \frac{2}{3} & 1 & \frac{2}{3} & 0 & \frac{1}{3} & \frac{8}{3} \\ \hline 0 & -\frac{2}{3} & 0 & -\frac{5}{3} & 0 & -\frac{1}{3} & -6\frac{2}{3} \end{array}\right)$$

All coefficients in the last row of the matrix are non-positive. We stop here.
We obtain that $x_1 = \frac{4}{3}$, $x_2 = 0$, $x_3 = \frac{8}{3}$. The maximum value of the objective function is $6\frac{2}{3}$.

(b)

$$\begin{array}{llcl} \textbf{min} & 4y_1 + 5y_2 & & \\ \textbf{s.t.} & y_1 \quad\; - 2y_3 & \geq & 1 \\ & y_1 + 3y_2 & \geq & 1 \\ & y_1 + y_2 + y_3 & \geq & 2 \\ & y_1 \geq 0,\ y_2 \geq 0,\ y_3 \geq 0 & & \end{array}$$

Question 3 (40 points). Let $\mathcal{N}(\mathcal{G}(\mathcal{V}, \mathcal{E}), s, t, c)$ be a flow network, where s is a source, t is a sink, $c : \mathcal{E} \to \mathbb{N}^+$ is a positive integer capacity function.

(a) Propose an efficient algorithm that, given an edge $e \in \mathcal{E}$, $c(e) > 0$, decides whether e belongs to **all** minimum cuts between s and t in $\mathcal{G}$.

(b) Propose an eficient algorithm that, given an edge $e \in \mathcal{E}$, $c(e) > 0$, decides whether e belongs to **some** minimum cut between s and t in $\mathcal{G}$.

In both parts of the question, prove the correctness of your solution and analyze its complexity.

Answer (a)

1. Run Dinitz algorithm to find the maximum flow between s and t. Denote the maximum flow by f_1.
2. Increase the capacity of the edge e by 1.
3. Run Dinitz algorithm to find the maximum flow between s and t. Denote the maximum flow by f_2.
4. The edge e belongs to all minimum cuts between s and t if and only if $f_2 = f_1 + 1$.

Quick explanation (this is not a proper proof): if e belongs to all minimum cuts, then by increasing its capacity by one we increased the capacity of all minimum cuts by one (and all non-minimum cuts had capacity $\geq f1 + 1$), and so $f_2 = f_1 + 1$. On the other hand, if there was a minimum cut that e does not belong to it, its capacity is still f_1.

(b)

1. Run Dinitz algorithm to find the maximum flow between s and t. Denote the maximum flow by f_1.
2. Decrease the capacity of the edge e by 1.
3. Run Dinitz algorithm to find the maximum flow between s and t. Denote the maximum flow by f_2.
4. The edge e belongs to some minimum cut between s and t if and only if $f_2 = f_1 - 1$.

Quick explanation (this is not a proper proof): if e belongs to some minimum cut, then by decreasing its capacity by one we decreased the capacity of this minimum cut by one, and so it is a minimum cut also in a new graph with $f_2 = f_1 - 1$.

On the other hand, if e does not belong to any minimum cut, then the capacity of any minimum cut, f_1, is not affected by the change in the capacity of e. And, the capacity of any cut that e belongs to was at least $f_1 + 1$ before the decrease.

Complexity of Dinitz algorithm (in both parts of the question) is $O(|V|^2|E|)$.

Question 4 (30 points).

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a finite undirected **bipartite** graph, $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cap \mathcal{B} = \varnothing$, and all edges in the graph have one endpoint in $\mathcal{A}$ and one endpoint in $\mathcal{B}$. Such a graph will be called **thin** if the degree of every vertex in $\mathcal{A}$ is exactly 2. A subset of vertices $\mathcal{S} \subseteq \mathcal{B}$ is called **a set of representatives** if every vertex $v \in \mathcal{A}$ has at least one neighbor in $\mathcal{S}$.

Give an efficient factor 2 approximation algorithm that finds the smallest set of representatives in the thin graph as above. Proof its correctness and approximation factor.

Answer Construct a new graph $\mathcal{H}$. The vertex set of $\mathcal{H}$ will be $\mathcal{B}$. The edge set of $\mathcal{H}$ will be defined as follows: if there was a vertex $v \in \mathcal{A}$ with two neighbors $u_1 \in \mathcal{B}$ and $u_2 \in \mathcal{B}$ in $\mathcal{G}$, then we draw an edge between u_1 and u_2 in $\mathcal{H}$.

We find a minimum vertex cover $\mathcal{S}$ for the graph $\mathcal{H}$. This can be done by an efficient algorithm with approximation factor 2 (shown in the class). The vertex cover $\mathcal{S}$ will be a set of representatives in $\mathcal{G}$.

Claim: a subset $\mathcal{S} \subseteq \mathcal{B}$ is a vertex cover in $\mathcal{H}$ if and only if $\mathcal{S}$ is a set of representatives in $\mathcal{G}$.

It follows from the claim that the sizes of the optimum vertex cover in $\mathcal{H}$ and the optimum set of representatives in $\mathcal{G}$ are equal. Thus, the approximation factor for the given problem follows.

Note: It is also possible to solve this problem by using reduction to Minimum Set Cover problem.