CMPUT 675 - Assignment #1
Due Sep 23 by 1pm

Fall 2016, University of Alberta

Pages: 6

Exercise 1)

Marks: 5

Let $\mathcal{G} = (L \cup R; E)$ be a k-regular bipartite graph. That is, $\deg(v) = k$ for each $v \in V$. Show that $\mathcal{G}$ has a perfect matching. Conclude that E can be partitioned into k perfect matchings in polynomial time.

Exercise 2)

Marks: 5

Let $\Phi = (X, \mathcal{C})$ be an instance of 3-SAT where X is a finite set of variables and $\mathcal{C}$ is a finite set of clauses, each of which involves exactly three distinct variables (i.e. a clause does not contain both a variable and its logical negation).

e.g. $X = \{x_1, x_2, x_3, x_4\}, \mathcal{C} = \{(x_1 \vee \overline{x_2} \vee x_4), (\overline{x_1} \vee \overline{x_3} \vee x_4), (x_2 \vee x_3 \vee x_4)\}$ is a 3-SAT instance with 4 variables and 3 clauses.

Further, suppose for each $x \in X$ there are at most three clauses in $\mathcal{C}$ that involve the variable x (i.e. at most three literals of the form x or $\overline{x}$ throughout the instances).

Show that there is an assignment $\tau : X \to \{\textbf{true}, \textbf{false}\}$ that satisfies all clauses in $\mathcal{C}$ and that we can compute one in polynomial time.

Exercise 3)

Marks: 5

Let $\mathcal{G} = (L \cup R; E)$ be a connected bipartite graph and say $n = |L| + |R|, m = |E|$. A vertex $v \in L \cup R$ is **mandatory** if v is matched in every maximum matching.

A simple algorithm for finding all mandatory vertices simply computes n maximum matchings in the graphs obtained from $\mathcal{G}$ by deleting a single vertex and checks the sizes of these matchings.

Give a faster algorithm for finding **all** mandatory vertices. For full marks, you should provide an algorithm that finds all mandatory vertices in $O(n \cdot m)$ time.

Bonus (1 marks)
Show that in every bipartite graph, at least one endpoint of each edge is mandatory.

Exercise 4)

Marks: 5

Let X be a finite set. A **partial ordering** of X is a relation $\prec$ between items in X satisfying the following three properties:

- **irreflexive**: $v \not\prec v$ for any $v \in X$,
- **antisymmetric**: $u \prec v \Rightarrow v \not\prec u$ for distinct $u, v \in X$, and
- **transitive**: $u \prec v$ and $v \prec w \Rightarrow u \prec w$ for any $u, v, w \in X$.

For example, such a partial ordering arises over the vertices of a directed, acyclic graph G when we say $u \prec v$ if $u \neq v$ and there is a path from u to v.

A **ordered sequence** is a set of items $S \subseteq X$ such that for any two distinct $u, v \in S$ we have either $u \prec v$ or $v \prec u$. Note the singleton set $\{v\}$ is an ordered sequence for any $v \in X$.

An **independent set** is a set of items $I \subseteq X$ such that for any two distinct $u, v \in I$ we have $u \not\prec v$ and $v \not\prec u$.

Call $\mathcal{C} \subseteq 2^X$ a *sequence cover* if each $S \in \mathcal{C}$ is an ordered sequence and $\cup_{S \in \mathcal{C}} S = X$.

1. Show that if X is finite then

$$\max_{I \text{ independent set}} |I| = \min_{\mathcal{C} \text{ sequence cover}} |\mathcal{C}|.$$

   That is, the size of a largest independent set equals the minimum number of ordered sequence required to form a sequence cover.

2. Show how to compute a largest independent set and a minimum sequence cover in polynomial time. You may suppose the partially ordered set is represented as a directed graph $\mathcal{G} = (X, E)$ where $uv \in E$ means $u \prec v$.

Hint: Consider the bipartite graph with a copy of X on each side and an edge uv with u on the left and v on the right if $u \prec v$.

Exercise 5)

Marks: 5

In this exercise, you will devise an algorithm that decides if it is possible for your favourite team to come first in the league at the end of a sequence of games.

You are given n teams where team i has an initial number of points $p_i \in \mathbb{Z}_{\geq 0}$. You are also given a sequence of pairs of teams $(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)$ where $1 \leq i_k < j_k \leq n$ for each $1 \leq k \leq m$. Each entry (i_k, j_k) represents a game to be played between teams i_k and j_k.

Each game (i_k, j_k) that has yet to be played will result in either a win for i_k or a win for j_k and the winning team gets exactly one point from that game (no ties). After all games are played, the teams will be ranked according to their final score q_i. Here, q_i is the sum of p_i and the number of games from the list that team i won.

Your favourite team is team 1. Give a polynomial-time algorithm that decides if it is possible, through some ways of assigning wins/losses to the games in the list, for $q_1 > q_i$ for all $2 \leq i \leq n$.

Example

If there are 4 teams with initial scores $2, 0, 0, 0$ and the remaining games are $(2, 3), (3, 4), (2, 4)$ then one way to guarantee your team wins is for 2 to beat 3, 3 to beat 4, and then 4 to beat 2 for final scores $2, 1, 1, 1$.

If there are 4 teams with initial scores $2, 2, 1, 2$ and the games to be played are $(1, 4), (2, 3), (3, 4)$ then it is impossible.

Exercise 6)

Marks: 8

Let $\mathcal{G} = (V, E)$ be a directed graph. For each edge $e \in E$, suppose we have a **lower bound** $\ell(e) \in \mathbb{R}_{\geq 0}$ and an **upper bound** $\mu(e) \in \mathbb{R}_{\geq 0}$.

A **fully-conservative flow** (**FCF**) is an assignment of values $c(e) \in \mathbb{R}$ to edges $e \in E$ such that

- $\ell(e) \leq c(e) \leq \mu(e)$ for each $e \in E$, and
- $c(\delta^{in}(v)) = c(\delta^{out}(v))$ for each $v \in V$.

Prove the following:

Theorem 1 *There is a* **FCF** $c : E \to \mathbb{R}$ *if and only if*

1. $\ell(e) \leq \mu(e)$ *for each* $e \in E$, *and*
2. $\ell(\delta^{in}(U)) \leq \mu(\delta^{out}(U))$ *for each* $\emptyset \subsetneq U \subsetneq V$.

Moreover, if $\ell(e) \in \mathbb{Z}_{\geq 0}$ *and* $\mu(e) \in \mathbb{Z}_{\geq 0}$ *for each* $e \in E$ *and conditions 1) and 2) are satisfied, then there is such a* **FCF** c *with* $c(e) \in \mathbb{Z}$ *for each edge* $e \in E$. *Finally, a circulation can be computed in polynomial time if these conditions are met.*

Breakdown

Note your task is do the following four things, ensure you address all for full marks.

- Show that 1) and 2) are necessary for a **FCF** to exist.
- Show that 1) and 2) are sufficient for a **FCF** to exist.
- Show that if 1) and 2) are satisfied and all lower and upper bounds are integral, then there is a **FCF** c with $c(e) \in \mathbb{Z}$ for each edge $e \in E$.
- Show that if 1) and 2) are satisfied, then such a **FCF** can be computed in polynomial time.

Hint: Try initially setting $c(e) = \ell(e)$ for each edge $e \in E$ and then using an algorithm from the lectures to try and correct the "imbalance" at each vertex.

Exercise 7)

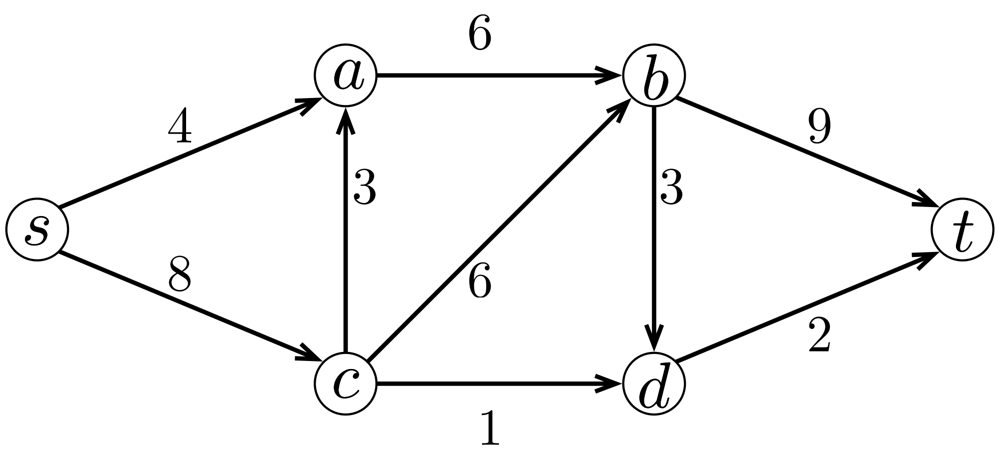
Marks: 5

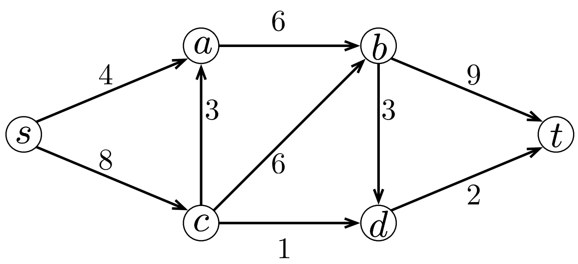
Compute a maximum $s-t$ flow in the following graph using the Goldberg-Tarjan PUSH-RELABEL algorithm. The number on an edge indicates the capacity of the edge.

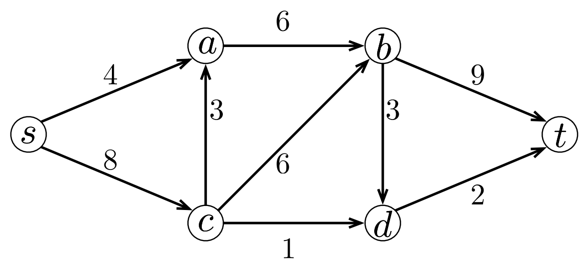
When selecting an active vertex v, you should pick one with highest distance label $\psi(v)$ among all active vertices. Apart from this criteria, your selection of active vertex and non-saturated edges in various steps of the algorithm may be arbitrary provided they are consistent with the description of the algorithm.

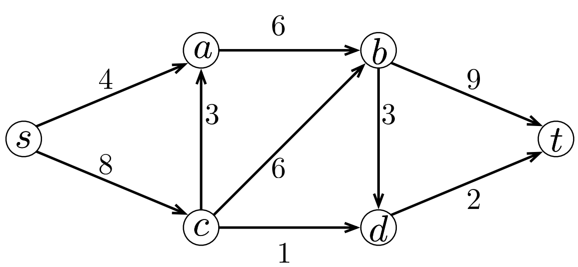
Carefully record each step of the algorithm. For convenience, I have included a page with extra figures that you may annotate (print out multiple copies if you want). **Clearly indicate** your choice of active vertex and nonsaturated edge in each step of the algorithm.

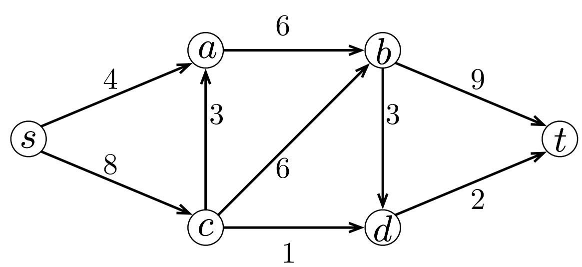
I suggest writing the distance label of the vertex right beside the vertex and the value of the flow across an edge beside the capacity of the edge, but it's up to you. Just explain your method clearly and **be clean**.

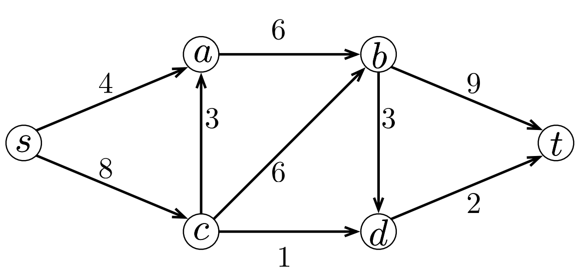


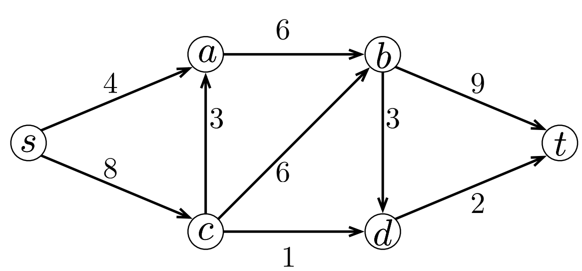

a
b
s
t
c
d
6
4
9
3
3
6
8
2
1

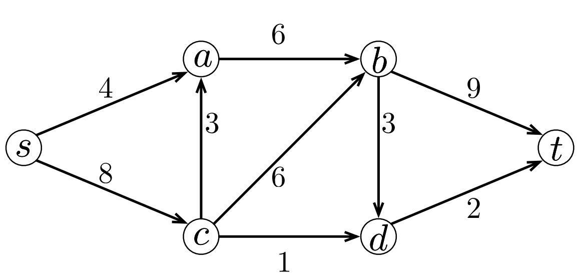

a
b
s
t
c
d
6
4
9
3
3
6
8
2
1

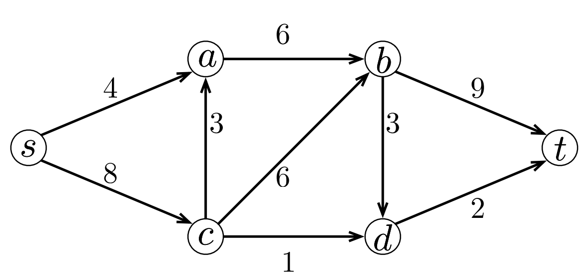

a
b
s
t
c
d
6
4
9
3
3
6
8
2
1


a
b
s
t
c
d
6
4
9
3
3
6
8
2
1


a
b
s
t
c
d
6
4
9
3
3
6
8
2
1


a
b
s
t
c
d
6
4
9
3
3
6
8
2
1


a
b
s
t
c
d
6
4
9
3
3
6
8
2
1


a
b
s
t
c
d
6
4
9
3
3
6
8
2
1