CMPUT 675 - Assignment #2
Due Oct 7 by 1pm

Fall 2016, University of Alberta

Pages: 5

Exercise 1)

Marks: 5

Let $\mathcal{G} = (V; E)$ be an undirected graph with capacities $\mu : E \to \mathbb{R}_{\geq 0}$. Suppose $n = |V|$ is even. An **odd-parity cut** is simply a cut $U \subseteq V$ with $|U|$ being odd. Call an odd-parity cut U a **minimum odd-parity cut** if every other odd-parity cut W has $\mu(\delta(W)) \geq \mu(\delta(U))$.

Let $\mathcal{T} = (V; F)$ be a Gomory-Hu tree for $\mathcal{G}$. Show that there is some edge $e \in F$ such that the fundamental cut of e in $\mathcal{T}$ is a minimum odd-parity cut.

Hint: you may want to break it down as follows.

- Show for any odd-parity cut X that there is some connected component $Y \subseteq X$ in the subgraph $\mathcal{T}[X] = (X, \{uv \in F : u, v \in X\})$ such that Y is an odd cut and $\delta_{\mathcal{T}}(Y) \subseteq \delta_{\mathcal{T}}(X)$.
- Then show there is some $st \in \delta_{\mathcal{T}}(Y)$ whose fundamental cut in $\mathcal{T}$ is an odd cut.
- Conclude that the capacity of this odd cut is at most $\mu(\delta_{\mathcal{G}}(X))$.

Exercise 2)

Marks: 5

Let $\mathcal{G} = (L \cup R; E)$ be a bipartite graph and $c : E \to \mathbb{R}$ edge costs. Give a polynomial-time algorithm for finding a matching $M \subseteq E$ of **maximum** possible cost.

For full marks, your algorithm should run in time $O(n \cdot m + n^2 \cdot \log n)$.

Exercise 3)

Marks: 8

Here you will give an incredibly simple randomized algorithm for finding a minimum-capacity cut in an undirected graph $\mathcal{G} = (V; E)$ with capacities $\mu : E \to \mathbb{R}_{\geq 0}$. Complete the following steps.

1. Let μ^* denote the capacity of a minimum-capacity cut. Show $\mu^* \leq \frac{2}{n}\mu(E)$ (**hint**: look at the cuts with a single vertex).
2. Let U^* denote a minimum-capacity cut (so $\mu^* = \mu(\delta(U^*))$). Argue that if one samples an edge e from the distribution that places probability $\frac{\mu(e')}{\mu(E)}$ on each $e' \in E$, then $\Pr[e \not\in \delta(U^*)] \geq \frac{n-2}{n}$.
3. Consider the algorithm that samples an edge as above, contracts the endpoints to a single vertex (keeping resulting parallel edges but discarding loops), and repeats with the resulting graph until only 2 vertices remain. Show that with probability at least $\frac{2}{n(n-1)}$, the set of edges that remain have capacity μ^*.

Note that if you repeat this algorithm $\frac{n(n-1)}{2}$ times, then the probability that the algorithm fails to find a minimum-capacity cut in any iteration is at most $\left(1 - \frac{2}{n(n-1)}\right)^{\frac{n(n-1)}{2}} \leq e^{-1}$ (you don't have to prove this, it's just an observation).

Exercise 4)

Marks: 5

Let $\mathcal{G} = (V; E)$ be an 3-regular, 2-edge connected undirected graph (not necessarily bipartite). That is, $\deg(v) = 3$ for each $v \in V$, $\mathcal{G}$ is connected, and it is impossible to create a disconnected graph by removing one edge from E. Show that $\mathcal{G}$ has a perfect matching.

Hint

First show for any $X \subseteq V$ that $|\delta(X)| \equiv |X| \bmod 2$. Then use Tutte's theorem.

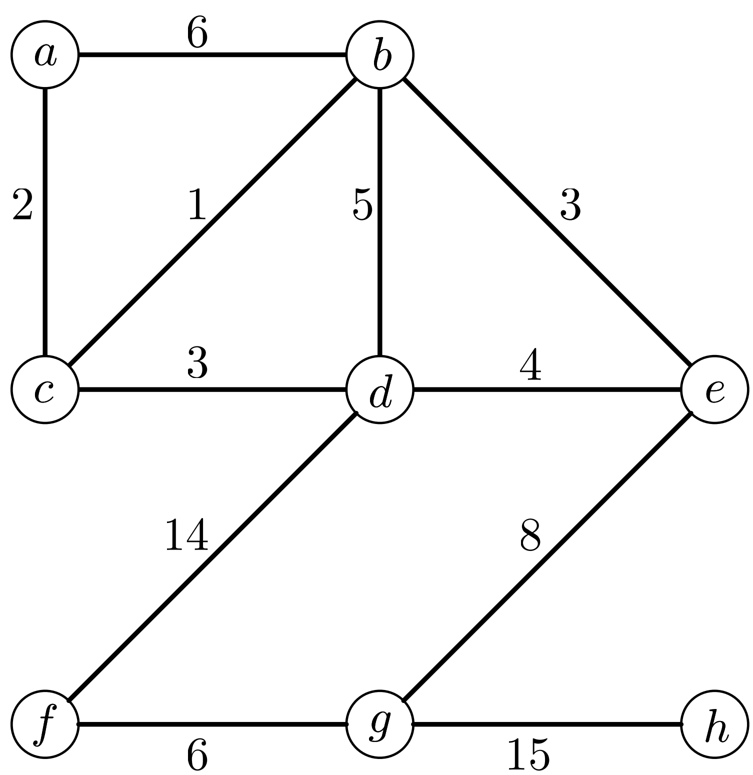
Bonus (1 mark): Prove that an undirected graph $\mathcal{H} = (V; E)$ is factor critical if and only if $\mathcal{H}$ is connected and $\nu(\mathcal{H}) = \nu(\mathcal{H} - v)$ for every $v \in V$.

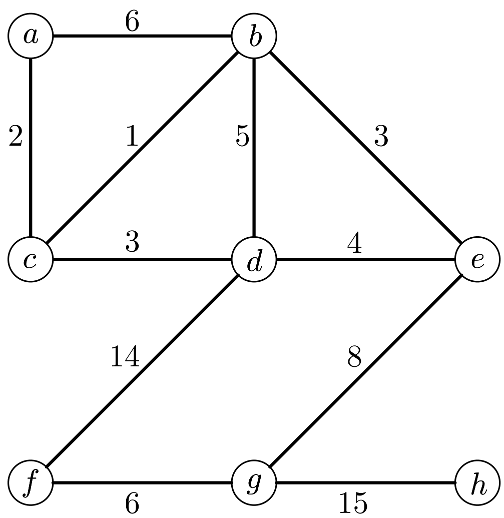
Exercise 5)

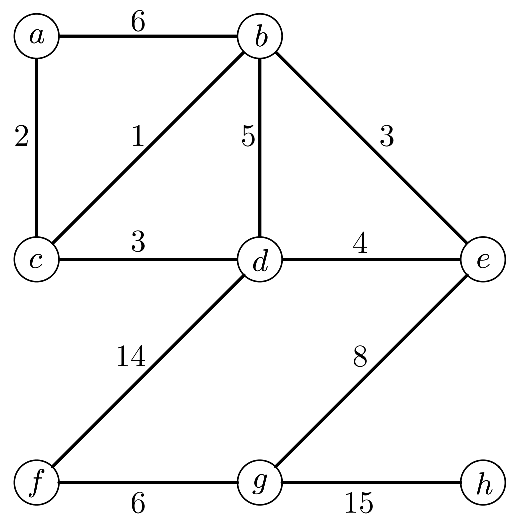
Marks: 5

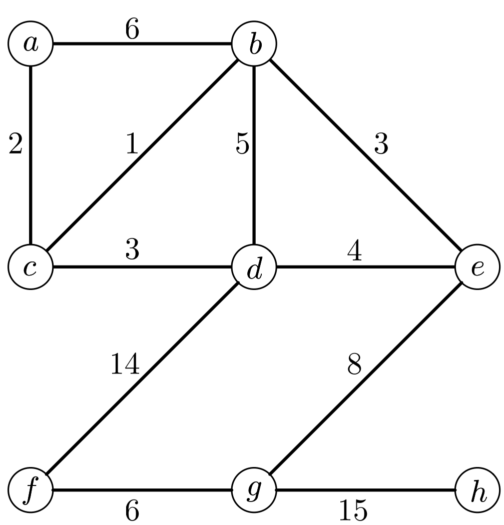
Construct a Gomory-Hu tree of the graph below using the algorithm from the lectures. Show your work: in each iteration indicate the vertex X of the tree and nodes $s, t \in X$ you are selecting to split, the corresponding minimum $s - t$ cut in G (either a picture or just listing the set of vertices in the cut), and depict the resulting tree.

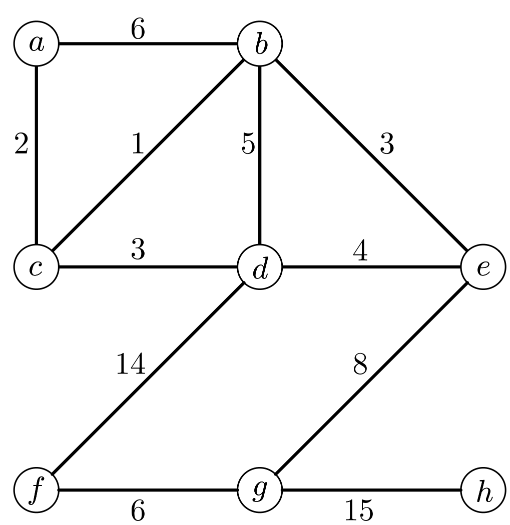
You do **not** need to show how you found the minimum $s - t$ cuts in $\mathcal{G}$.

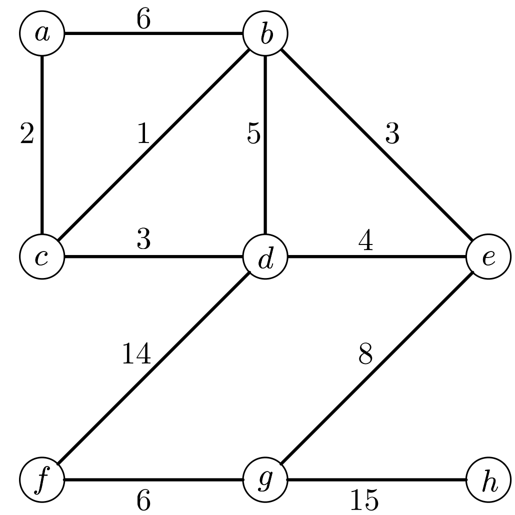
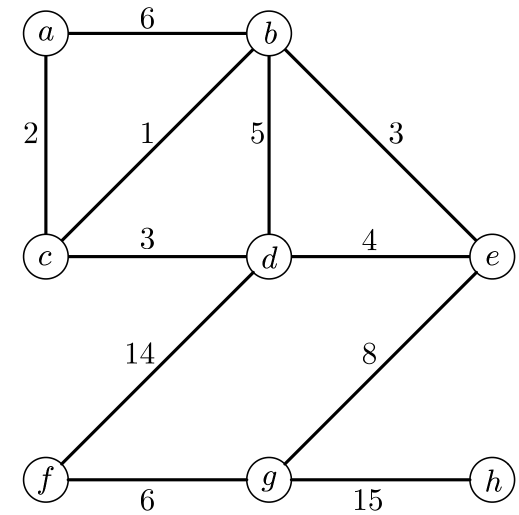
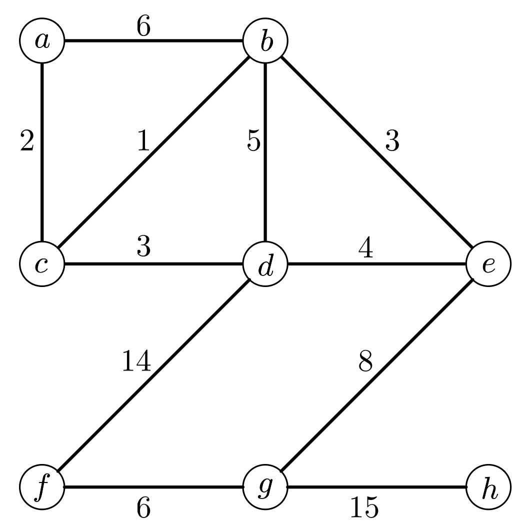
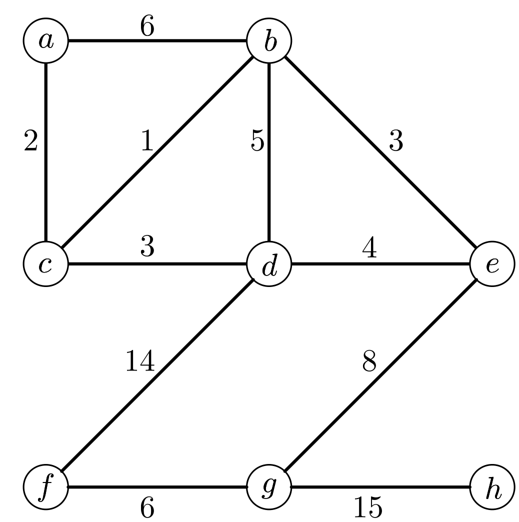



a
6
b
2
1
5
3
c
3
d
4
e
14
8
f
6
g
15
h


a
6
b
2
1
5
3
c
3
d
4
e
14
8
f
6
g
15
h


a
6
b
2
1
5
3
c
3
d
4
e
14
8
f
6
g
15
h


a
6
b
2
1
5
3
c
3
d
4
e
14
8
f
6
g
15
h


a
b
c
d
e
f
g
h
6
2
1
5
3
3
4
14
8
6
15

a
b
c
d
e
f
g
h
6
2
1
5
3
3
4
14
8
6
15

a
b
c
d
e
f
g
h
6
2
1
5
3
3
4
14
8
6
15

a
b
c
d
e
f
g
h
6
2
1
5
3
3
4
14
8
6
15