Gomory-Hu Trees

Let $G = (V, E)$ be an **undirected** graph and $u : E \to \mathbb{R}^+$ be capacities on edges.

Problem: Compute minimum s-t-cut for all pairs $(s, t) \in V^2$.

Simple solution: Run any maximum-flow algorithm $\binom{n}{2}$ times (it gives minimum cut too).

Better solution: Run $(n-1)$ times maximum-flow algorithm. Due to Gomory-Hu.

Denote the minimum capacity of an s-t-cut by λ_{st}.

1: Show that for any $i, j, k \in V(G)$, $\lambda_{ik} \geq \min\{\lambda_{ij}, \lambda_{jk}\}$.

Solution: Consider minimum i-k-cut $\delta(A)$, where $i \in A \subset V$. If $j \in A$, then A is also j-k-cut. Hence $\lambda_{ik} \geq \lambda_{jk}$. The other case is symmetrical.

A tree T is a **Gomory-Hu Tree** for (G, u) if $V(T) = V(G)$ and $\forall s, t \in V$

$$\lambda_{st} = \min_{e \in E(P_{st})} u(\delta_G(C_e)),$$

where P_{st} is the unique s-t-path in T, C_e is the set of vertices in the same connected component of $T - e$ as s and $\delta_G(C_e)$ is the cut defined by C_e in G.

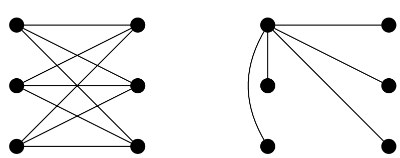
A Gomory-Hu tree is an object which quickly yields the values of minimum s-t-cuts. Note: the edges of T may not be edges of G.

2: Let (G, u) have a Gomory-Hu tree T. How to find the minimum cut from s to t, where s and t are vertices of G?

Solution:

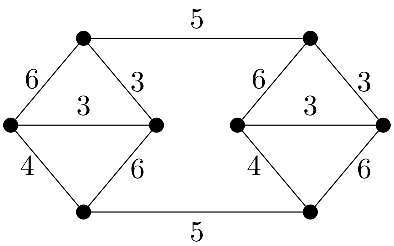
1. Find the path P_{st} from s to t in T.
2. For each edge e of P_{st}, deleting e, we are left with two components, C_e which contains s and the other component.
3. Calculate the capacity of that cut, which in our notation can be written as $u(\delta_G(C_e))$.
4. Return the minimum capacity found.

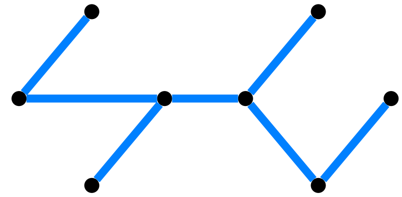
3: Example of G, where $u : E \to 1$ is a constant. Verify that T begin a star is indeed a Gomory-Hu three for (G, u).



Solution:

4: Find Gomory-Hu tree for the following graph with capacities on edges.



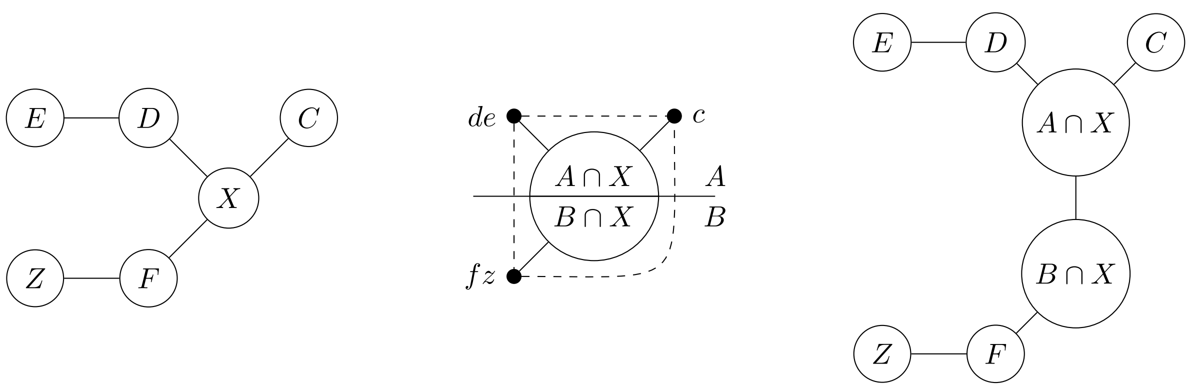


Solution: You probably failed to do it by hand. It is not even obvious that such structure might exist! And perhaps it might not be unique.

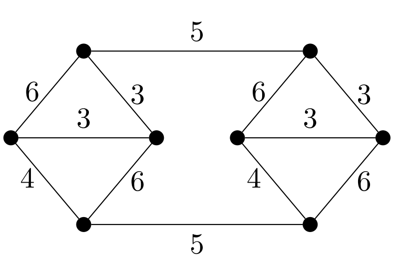
Algorithm:
We build a tree T, where each vertex of T corresponds to a *subset* of vertices of G. Eventually, all these subsets will have size one, i.e., T could be seen as on the same vertex set as G.

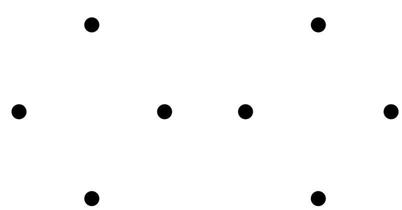
1. $T = (\{V\}, \emptyset)$
2. while exists $X \in V(T)$, where $|X| \geq 2$,
3. pick any s, t in X
4. make G' by contracting in G vertices of each connected component of $T - X$ into single vertex
5. in G' find a minimum s-t-cut $A \cup B = V(G')$
6. replace X in $V(T)$ by two adjacent vertices $\{A \cap X\}$ and $\{B \cap X\}$
7. if $y \in V(G') \setminus V(G)$ and $y \in A$, make the corresponding component of $T - X$ attached to $\{A \cap X\}$
8. analogously for $y \in V(G') \setminus V(G)$ and $y \in B$

Sketch of one iteration:

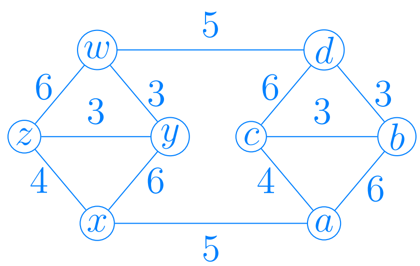


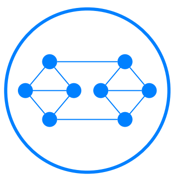
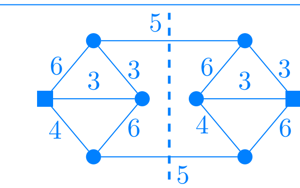
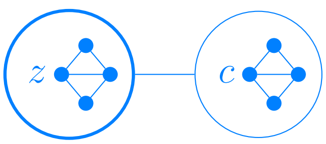
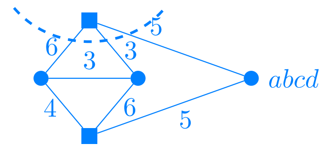
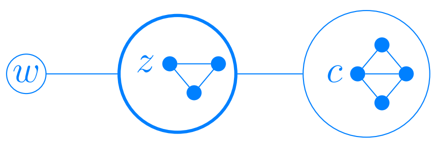
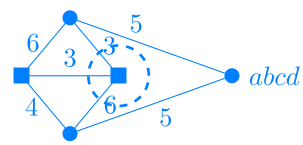
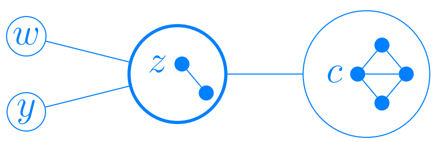
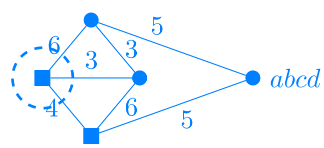
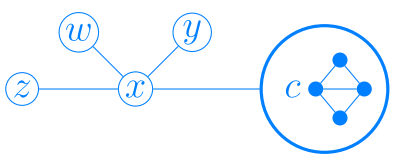
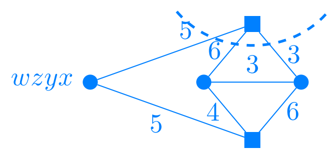
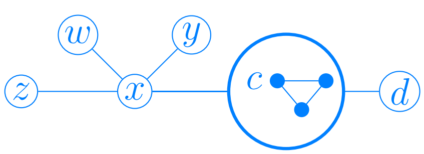
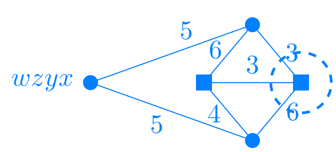
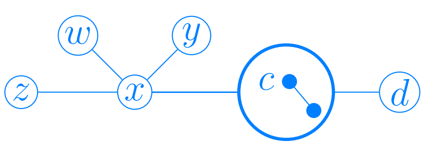
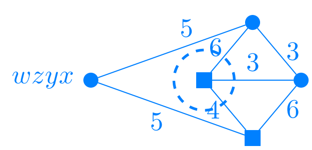
5: Run the algorithm on the following graph.

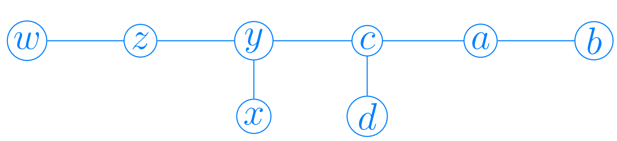




Solution: Lets start by giving all vertices a label. Then we give a table describing the run.



T and a selected node	G' and a minimum cut between squared vertices
	
	
	
	
	
	
	

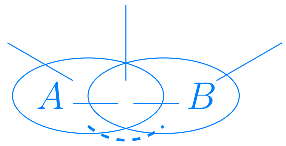
The final answer is  .

Note: It is helpful to write the cut values over the edges of the Gomory-Hu tree during each step.

6 *Cuts are submodular***:** That is, let $A, B \subset V$. Show that

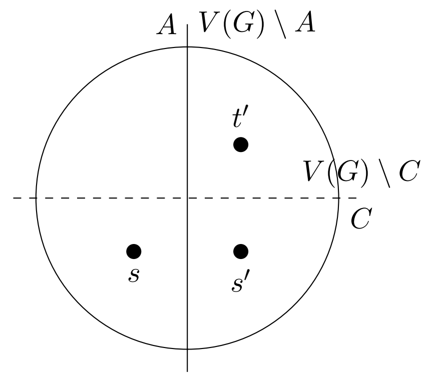
$$u(\delta(A \cup B)) + u(\delta(A \cap B)) \leq u(\delta(A)) + u(\delta(B)).$$

Solution: All edges on the left are counted on the right but edges going between $A \setminus B$ and $B \setminus A$ are missing on the left hand side. The dashed edges are missing.



7 *Algorithm creates optimal cuts***:** Let $s, t \in V$ and let $A \subset V$ such that $\delta(A)$ is a minimum s-t-cut. Let $s', t' \in V \setminus A$. Let (G', u') be obtained from G by contracting vertices of A into one vertex a'. Let $K \subset V(G')$ such that $\delta_{G'}(K \cup \{a'\})$ is a minimum s'-t'-cut in G'. Show that $\delta_G(K \cup A)$ is a minimum s'-t'-cut in G.

Proof beginning: Assume $\delta(C)$ is a minimum s'-t' cut in G. Show that $\delta(C \cup A)$ is also a minimum s'-t' cut in G. Without loss of generality $s \in A \cap C$. *Hint: Submodularity*



Solution: By submodularity, we have

$$u(\delta(A \cup C)) + u(\delta(A \cap C)) \leq u(\delta(A)) + u(\delta(C))$$

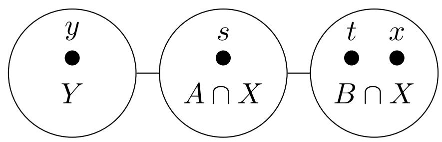
Observe that $\delta(A \cap C)$ is an s-t-cut. Hence $u(\delta(A)) \leq u(\delta(A \cap C))$. Therefore $u(\delta(A \cup C)) \leq u(\delta(C))$ and $\delta(A \cup C)$ is also minimum s'-t'-cut.

8: Let T be a tree during the run of algorithm. Let $e \in E(T)$ be any edge of T. Denote the endpoints of e by X and Y. Show that there are vertices $x \in X$ and $y \in Y$ such that e describes a minimum x-y cut.

The algorithm produces tree that works like Gomory-Hu tree at least for vertices adjacent in the tree.

Proof start: At the beginning of the algorithm (or after the first iteration, the observation is true. Let X and s, t be from step 3. in T, the new edge $A \cap X$ to $B \cap X$ is correct due to s, t. Edges not incident with X are also correct. Remaining are edges that used to be incident with X but now are incident with $A \cap X$ (or $B \cap X$).

Suppose the edge YX had vertices y and x. If Y is in the A part of s-t-cut and x is in the other part, the edge $Y, (X \cap A)$ needs to be verified to satisfy the conclusion.



Solution: We show $\lambda_{sy} = \lambda_{ys}$, which gives the desired that the edge Y–$A \cap X$ represents a minimum y-s-cut. We get

$$\lambda_{sy} \geq \min\{\lambda_{st}, \lambda_{tx}, \lambda_{xy}\}.$$

If we contract $B \cap X$ to one vertex, λ_{sy} does not change. Hence

$$\lambda_{sy} \geq \min\{\lambda_{st}, \lambda_{xy}\}.$$

Note that the A-B-cut separates x and y, hence $\lambda_{st} \geq \lambda_{xy}$ and we have

$$\lambda_{sy} \geq \lambda_{xy}.$$

On the other hand, the X-Y cut has size λ_{xy} and it is also s-y-cut. Hence

$$\lambda_{sy} \leq \lambda_{xy}$$

and the equality is proved.

9: Show that the tree produced by the algorithm is indeed a Gomory-Hu tree.

Solution: Let u, v be two vertices. Let $P_{uv} = (u = x_1, x_2, \ldots, x_n = v)$ be the path with endpoints u and v in T.

Observe

$$\lambda_{uv} \geq \min_{1 \leq i \leq n-1} \{\lambda_{x_i x_{i+1}}\}.$$

Since $\lambda_{x_i x_{i+1}}$ is obtained by edge $x_i x_{i+1}$ in the tree, we are done.