1 Dinitz algorithm

Find a maximum flow between \( s \) and \( t \) in the following network by using Dinitz algorithm:

![Diagram of the network](image)

Demonstrate the main steps in the algorithm. Show all minimum cuts. How many different minimum cuts can you find?

1.1 Dinitz algorithm

The first layered network is as follows

![Diagram of the first network](image)

Here we can push flow 2 on the path \( s \rightarrow a \rightarrow c \rightarrow t \) and flow 2 on \( s \rightarrow b \rightarrow d \rightarrow t \), giving us the following flow network:
This can be used to draw out the residual network in a straightforward manner. So we more directly to the layered networks:

Now we can add flow 1 on the path $s \rightarrow b \rightarrow d \rightarrow c \rightarrow t$ resulting in the following network:

Which gives us the following layered network:

where we can push flow 2, to get the following flow network:
However, in the next layered network we can only reach node $a$ from $s$ and there is no path to $t$. Hence, Dinitz algorithm terminates and the last illustration shows the maximum flow that has the value $F = 7$.

1.2 Minimum cuts

This network has three cuts with capacity 7:

- $S = \{s, a\}$
- $S = \{s, a, b\}$
- $S = \{s, a, b, c, d\}$

2 Combining minimum cuts

Let $\mathcal{N}(G(V, E), s, t, c)$ be a flow network. Consider two sets of vertices $S_1$ and $S_2$, where $\{s\} \subseteq S_1 \subseteq V \setminus \{t\}$ and $\{s\} \subseteq S_2 \subseteq V \setminus \{t\}$. It is known that both $(S_1 : \overline{S_1})$ and $(S_1 : \overline{S_2})$ are minimum cuts in $\mathcal{N}$. Define sets of vertices $T_1 = S_1 \cup S_2$ and $T_2 = S_1 \cap S_2$. Prove that both $(T_1 : \overline{T_1})$ and $(T_2 : \overline{T_2})$ are minimum cuts too.

Firstly, consider the basic properties of valid cuts $s \in T_1$ and $s \in T_2$ and $t \in \overline{T_1}$ and $t \in \overline{T_2}$. These clearly hold by definition and hence, $(T_1 : \overline{T_1})$ and $(T_2 : \overline{T_2})$ are valid cuts.

Secondly, we need to show that they are minimum cuts. We use the property that for a minimum cut $C : \overline{C}$ all edges in this cut are saturated and all edges in the reverse direction are empty. This is indeed true from the definition of the flow value

$$F = \sum_{e \in (C : \overline{C})} f(e) - \sum_{e \in (\overline{C} : C)} f(e) \leq \sum_{e \in (C : \overline{C})} c(e) - 0$$

where the final inequality is satisfied by equality for the minimum cut, and this can only hold if $\sum_{e \in (C : \overline{C})} f(e) = \sum_{e \in (C : \overline{C})} c(e)$ and $\sum_{e \in (\overline{C} : C)} f(e) = 0$ as we have non-negative flow values.
Consider $T_1 = S_1 \cup S_2$ then we have

$$T_2 = V \setminus T_1 = V \setminus S_1 \cup S_2 = V \setminus S_1 \cap V \setminus S_2 = \overline{S_1} \cap \overline{S_2}$$

using basic properties of set operations. All edges $e = (a, b)$ in $(T_1 : \overline{T_1})$ are saturated because they are either in $(S_1 : \overline{S_1})$ or $(S_2 : \overline{S_2})$. In more detail $a \in S_1 \cup S_2$ means that either $a \in S_1$ or $a \in S_2$ and from $b \in \overline{S_1} \cap \overline{S_2}$ we have $b \in \overline{S_1}$ and $b \in \overline{S_2}$. Hence, for $a \in S_1$ we have $(a, b) \in (S_1 : \overline{S_1})$ and for $a \in S_2$ we have $(a, b) \in (S_2 : \overline{S_2})$ and therefore this edge is saturated in the maximum flow. Similarly, all edges $e = (b, a)$ in $e \in (\overline{T_1} : T_1)$ must have zero flow $f(e) = 0$ because $a \in S_1 \cup S_2$ and $b \in \overline{S_1}$ and $\overline{S_2}$, hence depending on where $a$ really is we have $(b, a) \in (\overline{S_1} : S_1)$ or $(b, a) \in (\overline{S_2} : S_2)$ and therefore it has no flow because these cuts are minimal. Hence, $(T_1 : \overline{T_2})$ is a minimum cut.

The argument for $T_2 = S_1 \cap S_2$ is analogous to that of the previous claim. Note that here $T_2 = \overline{S_1} \cup \overline{S_2}$. Hence, for any $(b, a) \in (T_2 : \overline{T_2})$ we have that $b \in S_1$ and $b \in S_2$ and $a \in \overline{S_1}$ (and $(b, a) \in (S_1 : \overline{S_1})$ or $a \in \overline{S_2}$ (and $(b, a) \in (S_2 : \overline{S_2})$). Hence, $(b, a)$ is saturated as it belongs to one known minimum cut in the graph. For any $(a, b) \in (T_2 : \overline{T_2})$ we use the same cases to obtain that if $a \in \overline{S_1}$ then $(a, b) \in (\overline{S_1} : S_1)$ and therefore it has zero flow.

## 3 Properties of flows

Let $\mathcal{N}(G(V, E), s, t, c)$ be a network, where $G$ is a finite directed graph, and $c : E \rightarrow \mathbb{R}^+$ is a capacity function. For any two vertices $x, y \in V$, denote by $F_{x,y}$ the maximum flow in $\mathcal{N}$ when $x$ is the source and $y$ is the sink. Prove that for all vertices $u, v, w \in V$,

$$F_{u,v} \geq \min\{F_{u,w}, F_{w,v}\}.$$

From the max-flow min-cut theorem we know that if $F_{u,v}$ is a maximal flow between $u$ and $v$ then there also exists a minimal cut $(S : \overline{S})$ such that $u \in S$ and $v \in \overline{S}$ and the capacity $c(S : \overline{S})$ is minimum among all the cuts that satisfy the previous conditions. In addition, $F_{u,v} = c(S : \overline{S})$.

We have two possibilities to consider:

1. $w \in S$. Consider the flow $F_{w,v}$ where currently $(S : \overline{S})$ is a cut where the source is in $w \in S$ and $v \in \overline{S}$. However, this may not be the minimum cut between $w$ and $v$. The capacity of the minimum suitable cut $(S' : \overline{S'})$ between $v$ and $w$ would be equal to $F_{w,v} = c(S' : \overline{S'})$. Hence, by definition $F_{u,v} = c(S : \overline{S}) \geq c(S' : \overline{S'}) = F_{w,v}$. This implies $F_{u,v} \geq F_{w,v} \geq \min\{F_{w,v}, F_{u,v}\}$.

2. $w \in \overline{S}$. In this case the argument is similar to the previous case, but we consider the flow $F_{u,w}$. Now, $(S : \overline{S})$ is an interesting cut with the source $u \in S$ and the sink $w \in \overline{S}$ and it still has the same capacity $c(S : \overline{S})$. However, this is not necessarily a minimal cut between $u$ and $w$, hence, $F_{u,v} = c(S : \overline{S}) \geq F_{u,w} \geq \min\{F_{w,v}, F_{u,v}\}$.

Therefore, $\forall u, v, w \in V$ we have $F_{u,v} \geq \min\{F_{u,w}, F_{w,v}\}$. 

4
4 Minimum cuts

Let $\mathcal{N}(\mathcal{G}(V, E), s, t, c)$ be a flow network, where $s$ is a source, $t$ is a sink, $c : E \rightarrow \mathbb{Q}^+$ is a positive rational capacity function.

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

(b) Propose an efficient algorithm that, given an edge $e \in E$, $c(e) > 0$, 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 analyse its complexity.

4.1 All minimum cuts

Here we rely on the min-cut max-flow theorem and use the knowledge that the capacity of the minimum cut is equal to the capacity of the maximum flow.

1. Run Dinitz algorithm to find the maximum flow between $s$ and $t$. Denote the maximum flow by $f_1$. (First network.)

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 $f_2$. (Second network)

4. The edge $e$ belongs to all minimum cuts between $s$ and $t$ if and only if $f_2 > f_1$.

Quick explanation (this is not a proper proof): if $e$ belongs to all minimum cuts, then by increasing its capacity by 1 we increased the capacity of all minimum cuts and hence, the maximum flow has to increase. On the other hand, if there was a minimum cut such that $e$ does not belong to it, its capacity is still $f_1$ and the flow can not increase in the second network.

4.1.1 Proof

Edge $e$ belongs to all minimum cuts if and only if $f_2 > f_1$. Assume that $e$ belongs to all minimum cuts in the original network we need to show that $f_2 \geq f_1$. The capacity of these minimum cuts is $f_1$. In fact, all cuts that contain $e$ have their capacities increased by 1 in the second network, whereas all other cut capacities remain the same. Assume by contradiction that $f_2 \leq f_1$, then there exists some cut with capacity $f_2$ in the second network. However, this cut has the capacity $k \leq f_2$ also in the first network, because we only increased edge capacities. Note that $f_2 < f_1$ is impossible as we did not decrease the capacities of any edge. The only remaining possibility is $f_2 = f_1$ which contradicts the assumption that $e$ belongs to all minimum cuts because for $f_1 = f_2$ the capacity of the minimum cuts has to be the same in the first and second network, however we saw that if $e$ belongs to all minimum cuts in the first network, then the minimum cut of the second network has capacity $f_2 > f_1$. 
For the other direction assume that $f_2 > f_1$ and we need to show that $e$ belongs to all minimum cuts. By min-cut max-flow theorem we know that the minimum cut in the second network has larger capacity than the minimum cut in the first network. Assume, by contradiction, that $e$ does not belong to all minimum cuts. Consider a minimum cut $C$ of the first network that does not contain $e$. Now the capacity of this cut in the second network remains the same as in the first and is $c(C : \overline{C}) = f_1$. However, then we have $f_2 \leq f_1$ because each flow is at most as large as any cut. This is a contradiction with $f_2 > f_1$, therefore $e$ has to belong to all minimum cuts.

4.2 Some minimum cuts

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 $c(e)$ so that it becomes 0.

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 $f_2 = f_1 - c(e)$.

Quick explanation (this is not a full proof): if $e$ belongs to some minimum cut, then by decreasing its capacity by $c(e)$ we decreased the capacity of this minimum cut by $c(e)$, and so it is a minimum cut also in a new graph with $f_2 = f_1 - c(e)$. One could also decrease the capacity by any other value $d \leq c(e)$ to get the same result for condition $f_2 = f_1 - d$.

On the other hand, if $e$ does not belong to any minimum cut, then for any cut $C$ containing $e$, the capacity $c_1(C : \overline{C}) > f_1$ in the first network and in the second network the capacity $c_2(C : \overline{C}) = c_1(C : \overline{C}) - c(e) > f_1 - c(e)$. The only cuts where the capacities were decreased were the ones containing $e$. Hence, there are two possibilities: either the second network has the same max-flow as before $f_2 = f_1$ or some cut $C$ containing $e$ is the new minimum cut in the second network. For $f_1 = f_2$ clearly $e$ did not belong to any minimum cut. For $f_2 = c_2(C : \overline{C})$ we have already argued that then $f_2 > f_1 - c(e)$. Hence, the algorithm gives a correct result.

4.3 Complexity of both cases

We do some constant times operations and two flow computations in both algorithms. Hence, the complexity will fall down to the complexity of running a max-flow algorithm, for example Dinitz algorithm. Complexity of Dinitz algorithm is $O(|V|^2|E|)$ which is also the complexity of the proposed solution.