1. Let $f$ be a flow in $G$. Prove the following:

   A. For any $s, t$-cut $A, B$, $|f| = f(A, B)$.

   We will prove this by induction on the size of $A$. By definition,
   
   $|f| = f(\{s\}, V \setminus \{s\})$.

   Suppose we know that for any $s, t$-cut $A, B$ with $|A| = k$, $|f| = f(A, B)$. Consider an $s, t$-cut $A, B$ with $|A| = k + 1$. Let $x \in A$ and define $A' = A \setminus \{x\}$ and $B' = B \cup \{x\}$. Since $f$ is a flow, we know it satisfies $\sum_{v \in V \setminus \{x\}} f(x, v) = 0$ from which it follows that
   
   $\sum_{v \in B} f(x, v) = \sum_{u \in A'} f(u, x)$.

   Therefore,
   
   $f(A, B) = \sum_{u \in A'} \sum_{v \in B} f(u, v)$
   
   $= \sum_{u \in A'} f(u, v) + \sum_{v \in B} f(x, v)$
   
   $= \sum_{u \in A'} f(u, v) + \sum_{u \in A'} f(u, x)$
   
   $= \sum_{u \in A'} f(u, v)$
   
   $= f(A', B')$,

   and since $|A'| = k$, $|f| = f(A', B')$ by the inductive hypothesis, and we conclude that $|f| = f(A, B)$.

   B. Let $G_f$ be the residual graph. If $f^*$ is a max-flow in $G$, then the value of a max-flow in $G_f$ is $|f^*| - |f|$. 


Let \( f' = f^* - f \). Then \( f + f' = f + f^* - f = f^* \) is a max-flow in \( G \), so \( f' \) is a max-flow in \( G_f \) (follows from Lemma). Also from the Lemma, we have
\[
|f'| = |f^* - f| = |f^*| - |f|.
\]

2. Given a flow \( f \) on a directed graph \( G \) with positive edge capacities:

A. Show how to construct the residual graph \( G_f \) in \( O(m) \) time.

This is straightforward. For each edge \((u, v)\) in \( G \), compute the residual capacity \( r(u, v) = c(u, v) - f(u, v) \) and the reverse capacity \( r(v, u) = c(v, u) - f(v, u) \). For each edge \((u, v)\) such that \( r(u, v) > 0 \), add the edge to the residual graph \( G_f \) with capacity \( r(u, v) \). Do the same for \((v, u)\). This algorithm processes each of \( m \) edges in \( G \), doing constant-time work for each edge. Therefore the algorithm is \( O(m) \).

B. We described Dijkstra’s algorithm in class. Also, pseudocode is given in CLRS on page 658. Normally, Dijkstra’s algorithm minimizes weighted distance from the source, but we want to maximize bottleneck capacity from the source. To begin, rather than initializing \( v.d \) to \( \infty \) for each vertex \( v \), we will initialize \( v.d \) to \( r(s, v) \) and \( v.\pi \) to \( s \). Also, our solution set \( S \) will initially contain the source \( s \). Since we are maximizing the bottleneck capacity, replace the min-heap with a max-heap and call \texttt{Extract-Max} rather than \texttt{Extract-Min}. Finally, the function \texttt{RELAX}(u, v, r) must be changed so that if \( u.d + r(u, v) > v.d \) then we set \( v.d = u.d + r(u, v) \) and \( v.\pi = u \).

(26.1-7) The basic idea is pretty simple: we will construct \( G' \) by splitting each vertex \( v \) of \( G \) into two vertices \( v_1, v_2 \), joined by an edge of capacity \( l(v) \). All incoming edges of \( v \) are now incoming edges to \( v_1 \). All outgoing edges from \( v \) are now outgoing edges from \( v_2 \).

More formally, construct \( G' = (V', E') \) with capacity function \( c' \) as follows. For every \( v \in V \), create two vertices \( v_1, v_2 \) in \( V' \). Add an edge \((v_1, v_2) \in V' \) with \( c'(v_1, v_2) = l(v) \). For every edge \((u, v) \in E \), create an edge \((u_2, v_1) \in E' \) with capacity \( c'(u_2, v_1) = c(u, v) \). Make \( s_1 \) and \( t_2 \) the new source and sink vertices in \( G' \). Clearly, \(|V'| = 2|V|\) and \(|E'| = |E| + |V|\).

Let \( f \) be a flow in \( G \) that respects vertex capacities. Create a flow \( f' \in G' \) as follows. For each edge \((u, v) \in G \), let \( f'(u_2, v_1) = f(u, v) \). For each vertex \( u \in V \setminus \{s, t\} \), let \( f'(u_1, u_2) = \sum_{v \in V} f(u, v) \). Let \( f'(t_1, t_2) = \sum_{v \in V} f(v, t) \).

There is a one-to-one correspondence between flows that respect vertex capacities in \( G \) and flows in \( G' \). For the capacity constraint, every edge in \( G' \) of the form \((u_2, v_1)\) has a corresponding edge in \( G \) with a corresponding capacity and flow and thus satisfies the capacity constraint. For edges in \( E' \) of the form \((u_1, u_2)\), the capacities reflect the vertex capacities in \( G \). Therefore, for \( u \in V \setminus \{s, t\} \), we have \( f'(u_1, u_2) = \sum_{v \in V} f(u, v) \leq l(u) = c'(u_1, u_2) \). We also have \( f'(t_1, t_2) = \sum_{v \in V} f(v, t) \leq l(t) = c'(t_1, t_2) \). Note that this constraint also enforces the vertex capacities in \( G \).
Now, we prove flow conservation. By construction, every vertex of the form \( u_1 \in G' \) has exactly one outgoing edge \((u_1, u_2)\) and every incoming edge to \( u_1 \) corresponds to an incoming edge of \( u \in G \). Thus, for all vertices \( u \in V \setminus \{s, t\} \), the incoming flow to \( u_1 \) is given by

\[
\sum_{v \in V'} f'(v, u_1) = \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) = f'(u_1, u_2) = f'(u_1, u_2),
\]

and this last expression is just the outgoing flow from \( u_1 \).

For \( t_1 \), the incoming flow is given by

\[
\sum_{v \in V'} f'(v, t_1) = \sum_{v \in V} f(v, t) = f'(t_1, t_2),
\]

and this last expression is the outgoing flow.

Vertices of the form \( u_2 \) have exactly one incoming edge \((u_1, u_2)\), and every outgoing edge of \( u_2 \) corresponds to an outgoing edge of \( u \in G \). Thus for \( u_2 \neq t_2 \), the incoming flow is

\[
f'(u_1, u_2) = \sum_{v \in V} f(u, v) = \sum_{v \in V'} f'(u_2, v),
\]

which is equal to the outgoing flow.

Lastly, we prove that \(|f'| = |f|\):

\[
|f'| = \sum_{v \in V'} f'(s_1, v) = f'(s_1, s_2) = \sum_{v \in V} f(s, v) = |f|
\]