4(a)

We will modify Edmonds-Karp algorithm to get a polynomial time algorithm for the given problem. Given a network G, and a valid flow f on G, define a new residual graph G_f as follows: First, define the residual weights as

$$w_f(u,v) = \begin{cases} f(u,v) - w(u,v) & \text{if } (u,v) \in E \\ \infty & \text{if } (v,u) \in E \\ -1 & \text{otherwise} \end{cases}$$

Here, we are assuming that $(u,v) \in E \Leftrightarrow (v,u) \notin E$.

Now, define the residual graph $G_f = (V, E_f)$,
where $E_f = \{(u,v) \in V \times V | w_f(u,v) > 0\}$

Define a *decrementing path* to be a simple path from s to t in G_f. Now, given any *decrementing path*, say p, we can decrease the net flow on this path by an amount: $w_f(p) = \min\{w_f(u,v) | (u,v) \in p\}$, thus reducing the flow by $w_f(p)$.

Initial flow can be setup using DFS repeatedly (Note that initial flow is not 0, as was the case in the original max-flow min-cut case). It can be checked that setting up the initial flow takes polytime. Use Edmonds-Karp to decrease the flow along *decrementing paths* The proof of the termination of this algorithm is similar to that of Edmonds-Karp.

4(b)

It is not hard to find a counter example for min-flow = max-cut. Now, define a *forward cut* to be a cut $c(S,T)$ such that $\nexists t_1 \in T, t_2 \in S$ with $(t_1, t_2) \in E$

Analogue of max-flow min-cut theorem

The following are equivalent:

1. f is a minimum flow.
2. The residual network G_f contains no *decrementing path*.
3. $|f| = c(S,T)$ for some *forward cut* $c(S,T)$ of G.

   The proof of this proceeds along similar lines to that of the proof of max-flow min-cut theorem as given in CLRS.

   Note that such a cut does not exist $\Leftrightarrow$ then there is no minimum flow.