CS499 Homework 10 (First Draft)

Intersteller

Exercise 10.1

Since

$$\sum_{v \in V} f(s,v) = \sum_{v \in V \setminus S} f(s,v) + \sum_{v \in S} f(s,v)$$

we only need to prove that

$$\sum_{v \in S} f(s,v) = \sum_{u \in S-s,\ v \in V \setminus S} f(u,v)$$

Since

$$\sum_{v \in S} f(s,v) = -\sum_{v \in S-s} f(v,s)$$

we only need to prove that

$$\sum_{u \in S-s,\ v \in s+V \setminus S} f(u,v) = 0$$

It is obvious to see that

$$\sum_{u \in S-s,\ v \in S-s} f(u,v) = 0$$

So, we only need to prove that

$$\sum_{u \in S-s,\ v \in s+V \setminus S} f(u,v) + \sum_{u \in S-s,\ v \in S-s} f(u,v) = \sum_{u \in S-s,\ v \in V} f(u,v) = 0$$

According to the defination,

$$\sum_{u \in S-s,\ v \in V} f(u,v) = 0$$

Done.

Exercise 10.2

Define the minimum cut between i and j as minCut(i,j). According to the Max Flow Min Cut Theorem, minCut(s,r)≥k, minCut(r,t)≥k. Obviously, minCut(s,t)≥min{minCut(s,r),minCut(r,t)}≥k, which means there is a flow from s to r of value k.

Exercise 10.4

Suppose there is a s-t-path in G that has less that k edges. Then, at least one edge in the path moves more than one level forward, which contradicts Definition 10.3. So, $dist(s,t) \geq k$.

Exercise 10.6

Obviously, $(1) s \in V_0$ and $(2) t \in V_k$ are satisfied, we consider condition (3). Since (G, s, t, c) is a flow network and $V_0, V_1, ..., V_k$ is an optimal layering, every edge in G moves at most one level forward and $dist_G(s,t) = k$. And we denote the path p as $s \to v_1 \to v_2 \to ... \to v_{k-1} \to t$. Then we can get $s \in V_0$, $v_1 \in V_1$, ... ,$v_i \in V_i$, ... ,$v_{k-1} \in V_{k-1}$ and $t \in V_k$. Each edge is between two adjacent layerings. Since residual network G_f only add a reverse edge in each edge in p, these additional edge is also between two adjacent layerings. So $V_0, V_1, ..., V_k$ satisfy condition 3 and it is a layering of (G_f, s, t, c_f).

Exercise 10.8

We consider each while-loop of EK algorithm. In every iteration EK algorithm choose p to be a shortest s-t-path in G_f. And we denote the path p as $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_{k-1} \rightarrow v_k$ and $v_0 = s \in V_0$, $v_1 \in V_1$, ... ,$v_i \in V_i$, ... ,$v_{k-1} \in V_{k-1}$ and $v_k = t \in V_k$. Then EK algorithm routes c_{min} flow along p. So in G there $\exists(v_i, v_{i+1}) \in p$ where $i \in 0, 1, 2..., k$, $c_f(v_i, v_{i+1}) = c_{min}$ and in G_f, $c_f(v_i, v_{i+1}) = 0$ and $c_f(v_{i+1}, v_i) = c_{min}$. Obviously, after that $c_f(v_i, v_{i+1})$ is always 0 if $V_0, V_1, ..., V_k$ is still an optimal layering, otherwise there is a feasible s-t-path from v_{i+1} to v_i which is impossible. Therefore in every iteration, the total number of edges, which are from V_i to $V_{i+1}(\forall i \in 0, 1, 2, .., k)$ and in feasible s-t-path($dist(s, t) = k$), will minus at least one. Obviously these edges are less than or equal to m. So after m iteration, there no feasible s-t-path which $dist(s, t) = k$ and $dist(s, t)$ will be large than k. Therefore after at most m iterations of the while-loop, $V_0, V_1, ..., V_k$ ceases to be an optimal layering.

Exercise 10.9

proof According to **Exercise 10.8**, a particular layering is no more optimal after at moat m iterations. Since a layering is at least 1-layering and at most n-latering, after at most $m * n$ iterations, there is no optimal layering, which means there is no s-t-path, the algorithm terminates.

Exercise 10.10

proof According to **Exercise 10.9**, the Edmonds-Karp algorithm terminates after nm iterations of the while-loop, which is to say, we can get the max flow f after finite steps by Edmonds-Karp algorithm.

Question