Algorithms, 2023 Fall, Homework 10
(Due: Dec 6)

November 27, 2023

Network flow can be a tricky topic, as many seemingly unrelated stuffs are in fact connected to flows, via nontrivial transformations.

Problem 1 : @ KT book pp 415 problem 3.

Problem 2 : KT book pp 419 problem 9.

Problem 3 : A vertex cover is a set of vertices such that each edge is incident on at least one vertex in the set. An independent set is a set of vertices such that no edge joins two vertices in the set (i.e., no two vertices in the set are adjacent.) Prove the following statements:

1. In any bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.

2. In any bipartite graph, the number of vertices in a maximum independent set equals the total number of vertices minus the number of edges in a maximum matching.

Problem 4 : A b-regular graph is a graph such that the degree of each node is b. Show that a b-regular bipartite graph can be decomposed into b perfect matchings. (hint: use Hall's Theorem)

Problem 5 : $\mathcal{S}$ is a family of subsets of U. We say $\mathcal{S}$ is a *Laminar* family if for any two sets $S, T \in \mathcal{S}$, we have exactly one of the following holds:

1. $S \subset T$;

2. $T \subset S$;

3. $S \cap T = \emptyset$;

Now, we are given a set of elements U and a family $\mathcal{F}$ of subsets of U. $\mathcal{F}$ is the union of two Laminar families $\mathcal{S}_1$ and $\mathcal{S}_2$. For each $S \in \mathcal{F}$, there is an associated positive integer b_S. We would like to select a set $K \subseteq U$ of elements with the maximum cardinality such that $|K \cap S| \leq b_S$ for all $S \in \mathcal{F}$. Design a polynomial algorithm for this problem.

Problem 6 : Recall a function $f : 2^U \to \mathbb{R}$ is submodular if

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T), \;\; \forall S, T \subseteq U.$$

Consider an undirected graph $G(V, E)$. Let $\delta(S)$ be the total capacity of the cut defined by $(S, V - S)$, for $S \subseteq V$. Prove that $\delta(S)$ is a submodular function.

Problem 7 : Compute the polynomial $p(x) = \prod_{i=1}^{n}(a_i x + b_i)$, i.e., compute the coefficients of $p(x)$. Your algorithm should run faster than $O(n^2)$. Try to design an algorithm that runs as fast as possible.

Bonus problems: Bonus problems are problems that may be (may be not) more challenging than other problems. You do not lose points if you do not answer bonus problems. But you will get extra points if you get correct answers for such problems. **To make your life easier, we give you two weeks for the bonus problems**).

Problem 1 : We can improve the running time bound of FORD-FULKERSON by finding the shortest augmenting path (in terms of the number of edges). This is the Edmonds-Karp algorithm. We now prove that the Edmonds-Karp algorithm runs in $O(VE^2)$ time. Essentially, we need to show the number of iterations is bounded by $O(EV)$.

First, let us show this fact: The shortest s-v distance in G_f is monotonically increasing with iterations for any $v \in V$. (hint: suppose augmenting flow along a shortest path makes the s-v distance decrease. Try to derive a contradiction.)

Second, we define an edge e to be a *critical edge* when the residual capacity of e is the smallest in the shortest s-t path P in G_f (so after flow augmentation along path P, e will be saturated). Show that during the execution of the algorithm, an edge (u, v) becomes a critical edge at most $|V|/2$ times. (hint: When (u, v) becomes critical, it will disappear in the residual graph after this iteration. So when it can be critical again? it must happen after we push some flow along (v, u) later. Think about how the distances from s to u and v change during this process. In particular, show the s-u distance increases by at least 2 when u becomes critical again.)

I have provided enough hints. Please write down all details of the analysis.