Algorithm Design and Analysis
Assignment 5
Deadline: Jun 2, 2023

1. (35 points) **[Common System of Distinct Representatives]** Given a ground set $U = \{1, \ldots, n\}$ and a collection of k subsets $\mathcal{A} = \{A_1, \ldots, A_k\}$, a *system of distinct representatives* of $\mathcal{A}$ is a "representative" collection T of distinct elements from the sets in $\mathcal{A}$. Specifically, we have $|T| = k$, and the k *distinct* elements in T can be ordered as $u_1, \ldots, u_k$ such that $u_i \in A_i$ for each $i = 1, \ldots, k$. For example, $\{A_1 = \{2, 8\}, A_2 = \{8\}, A_3 = \{4, 5\}, A_4 = \{2, 4, 8\}\}$ has a system of distinct representatives $\{2, 4, 5, 8\}$ where $2 \in A_1, 4 \in A_4, 5 \in A_3, 8 \in A_2$, while $\{A_1 = \{2, 8\}, A_2 = \{8\}, A_3 = \{4, 8\}, A_4 = \{2, 4, 8\}\}$ does not have a system of distinct representatives.

   (a) (15 points) Design a polynomial time algorithm to decide if $\mathcal{A}$ has a system of distinct representatives.

   (b) (20 points) Given a ground set $U = \{1, \ldots, n\}$ and two collections of k subsets $\mathcal{A} = \{A_1, \ldots, A_k\}$ and $\mathcal{B} = \{B_1, \ldots, B_k\}$, a *common system of distinct representatives* is a collection T of k elements that is a system of distinct representatives of both $\mathcal{A}$ and $\mathcal{B}$. Design a polynomial time algorithm to decide if $\mathcal{A}$ and $\mathcal{B}$ have a common system of distinct representatives.

   For each part, prove the correctness of your algorithm, and analyze its time complexity.

2. (35 points) Consider the maximum flow problem $(G = (V, E), s, t, c)$ on graphs where the capacities for all edges are 1: $c(e) = 1$ for each $e \in E$. You can assume there is no pair of anti-parallel edges: for each pair of vertices $u, v \in V$, we cannot have both $(u, v) \in E$ and $(v, u) \in E$. You can also assume every vertex is reachable from s.

   (a) (20 points) Prove that Dinic's algorithm runs in $O(|E|^{3/2})$ time.

   (b) (15 points) Prove that Dinic's algorithm runs in $O(|V|^{2/3} \cdot |E|)$ time. (Hint: Let f be the flow after $2|V|^{2/3}$ iterations of the algorithm. Let D_i be the set of vertices at distance i from s in the residual network G^f. Prove that there exists i such that $|D_i \cup D_{i+1}| \leq |V|^{1/3}$.)

3. (35 points) In this question, we will prove König-Egerváry Theorem, which states that, in any bipartite graph, the size of the maximum matching equals to the size of the minimum vertex cover. Let $G = (V, E)$ be a bipartite graph.

   (a) (5 points) Explain that the following is an LP-relaxation for the maximum matching problem.

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} x_e & \\ \text{subject to} \quad & \sum_{e: e=(u,v)} x_e \leq 1 & (\forall v \in V) \\ & x_e \geq 0 & (\forall e \in E) \end{aligned}$$

   (b) (5 points) Write down the dual of the above linear program, and justify that the dual program is an LP-relaxation to the minimum vertex cover problem.

   (c) (10 points) Show by induction that the *incident matrix* of a bipartite graph is totally unimodular. (Given an undirected graph $G = (V, E)$, the incident matrix A is a $|V| \times |E|$ zero-one matrix where $a_{ij} = 1$ if and only if the i-th vertex and the j-th edge are incident.)

   (d) (10 points) Use results in (a), (b) and (c) to prove König-Egerváry Theorem.

   (e) (5 points) Give a counterexample to show that the claim in König-Egerváry Theorem fails if the graph is not bipartite.

4. (35 points) The network flow problem only restricts the capacity of each edge. Consider the following variant, each edge e does not only have a capacity c_e, but also a demand d_e. Finally, the edge should have flow $d_e \leq f_e \leq c_e$. Please find out how to solve the following problems by reducing them to the original max flow problem.

   (a) (20 points) In the original network flow problem, finding a feasible flow is easy. (a zero flow is surely feasible.) However, whether there exists a feasible flow is not straightforward in this new variant. How to determine the existence of a feasible flow by using the original max flow algorithm? (i.e., each f_e should satisfy $d_e \leq f_e \leq c_e$ and the flow conservation constraint should hold at all vertices other than s and t.)

   (b) (15 points) Based on the previous part, can we further find a maximum feasible flow? Please also use the original max flow algorithm.

5. How long does it take you to finish the assignment (including thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Write down their names here.