COMS 511 - Homework 5

Due: March 3 11:59 PM

Guidelines

- When proofs are required, you should make them both clear and rigorous. Do not hand-waive.
- Your assignment needs to be submitted via Canvas.
  - You **must** type your solutions. Please submit a PDF version.
  - Please make sure that the file you submit is not corrupted and that its size is reasonable (e.g., roughly at most 10-11 MB).

    If we cannot open your file, your homework will not be graded.
- The following are examples of activities that are prohibited:
  - Sharing solutions or fragments of solutions (e.g., via email, whiteboard, handwritten, or printed copies).
  - Post solutions or fragments of solutions in a location accessible to others.
  - Using solutions or fragments of solutions provided by other students (including students who had taken the course in the past).
  - Using solutions or solution fragments obtained on the Internet or from solution manuals for textbooks.
  - Using material from textbooks, reference books, or research articles without properly acknowledging and citing the source.
- Concerns about grading should be expressed within one week of returning the homework.
- **No late homework is accepted** with the exception of at most one late submission up to 12 hours late.

Problems

Problem 1. (50 points) Imagine living in an off-campus cooperative apartment with $n-1$ other people. Each of you has to cook dinner for the entire group exactly once over the next n nights. However, deciding who will cook each night is difficult due to everyone's scheduling constraints. Denote all people in the apartment by $\{p_1, p_2, ..., p_n\}$, and denote the next n nights by $\{d_1, d_2, ..., d_n\}$. For each person p_i, there is a set of nights $S_i \subseteq \{d_1, d_2, ..., d_n\}$ where he/she cannot cook. A dinner schedule is *feasible* if each person in the apartment cooks on exactly one night, someone cooks on each night, and if p_i cooks on the night d_j, then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not and formally analyze the runtime.

Problem 2. (50 points) Consider a bipartite graph $G = (V_1 \cup V_2, E)$, where $|V_1| = |V_2| = n$. A *perfect matching* in a graph G is a matching in which every vertex of G appears exactly once, that is, a matching of size exactly n. Given a bipartite graph G, describe an algorithm to determine if G has a perfect matching by a reduction to the max-flow problem. Formally analyze the runtime.