15-451/651 Algorithm Design & Analysis, Spring 2026

Homework #5

Homework #5, Due: Sunday, Mar 15 at 9:00pm

Guidelines:

1. This homework has three problems, all graded on completion.
2. In all cases the write-up you submit must be written by you alone. You may not share your written solutions with each other.
3. You can discuss the problem as a group, or consult the internet (in fact, references are provided directly when possible). However, you then must write up your final solutions independently in your own words.
4. When asked to "give an algorithm" or "show a runtime", you should:
   (a) describe your algorithm in English **OR** pseudocode,
   (b) provide a short argument for correctness and running time,
   (c) work with arbitrary, unknown, inputs, instead of on a single instance/example.

Submission: Submit to `gradescope` using join code that you received by email: please contact us if you did not find this code, or have trouble joining the course.

Your solutions **must** be typeset, **not handwritten**, and submitted as a PDF file. You should use US letter-size paper, a font size no smaller than 10pt, margins no smaller than 1in.

1. (**, disjoint short paths) Let G be a directed graph with m edges whose edges have positive lengths. Let s be a source vertex and let $v_1, v_2, \ldots, v_k$ be other vertices given as input. Your goal is to return edge-disjoint paths $P_1, P_2, \ldots, P_k$, where P_i is from $s \to v_i$, and minimize the total sum of their lengths. Give an algorithm to do this in time at most $O(m^{10})$.

   Can you give a simpler algorithm if the paths are not forced to be edge-disjoint?

   Hint. You should reduce to something discussed in class.

2. (**, multicommodity flow) Let G be a directed graph, and let $(s_1, t_1), \ldots, (s_k, t_k)$ be pairs of sources and sinks. We want to send one unit of flow for each pair $s_i \to t_i$ (**Important: the flows need not be integral.**) in a way that minimizes the maximum congestion on any edge. Formally, let $f^{(1)}, \ldots, f^{(k)}$ be the flows. Then the congestion of edge e is defined as

$$\text{cong}_e = \sum_{i=1}^{k} f_e^{(i)},$$

   where $f_e^{(i)}$ denotes the amount of flow on edge e in $f^{(i)}$. You want to find flows $f^{(i)}$, each routing one unit from s_i to t_i, that minimize $\max_e \text{cong}_e$.

   Write a linear program that captures this problem.

Think about the question in this paragraph yourself; you do not need to submit anything for this part. You should also try to capture this multicommodity flow problem as a maximum flow / minimum-cut problem, or as a min-cost flow problem. What goes wrong? Why does your attempted reduction fail?

3. (***, Hall's Lemma modified / Tutte-Berge formula on Bipartite Graphs) Let G be a bipartite graph with bipartition $V = L \cup R$, where L is the set of left vertices and R is the set of right vertices. Let $a = |L|$ and $b = |R|$ (L has a vertices and R has b vertices). Define a 2-matching of G to be a subset of edges $F \subseteq E(G)$ such that:

   (a) Every vertex in L is incident to at most 2 edges of F, and

   (b) Every vertex in R is incident to at most one edge of F.

   Prove that the maximum number of edges in a 2-matching is given by the following formula:

$$2a - \max_{S \subseteq L} (2|S| - |N(S)|),$$

   where $N(S) \subseteq R$ is the set of neighbors in R of the vertices in S.

   Hint. Set up the computation of a 2-matching as a flow problem. Use the max-flow min-cut theorem, and use the structure of the minimum cut to build the set S.