CS 473 (Spring 2025)
Homework 8 (due Apr 3 Thu 10am)

Instructions: As in previous homeworks.

Problem 7.1: Let $G = (V, E)$ be an unweighted graph with n vertices and m edges ($m \geq n$). A *loop* of length $\ell \geq 2$ refers to a walk $\langle v_1, v_2, \ldots, v_\ell, v_1 \rangle$ such that $v_1, v_2, \ldots, v_\ell$ are all distinct and $v_1v_2, v_2v_3, \ldots, v_\ell v_1 \in E$. Note that a loop is the same as a (simple) cycle, except for the case of $\ell = 2$ (usually $\langle v_1, v_2, v_1 \rangle$ is not considered a cycle, but is a loop of length 2 if $v_1v_2 \in E$). We study the following problem: find a vertex-disjoint collection of loops and paths, maximizing their total length.

(a) (20 pts) Show that there always exists an optimal solution using only loops of length 2 or odd lengths, and no paths. (So, we can redefine the problem without paths and the problem is the same.)

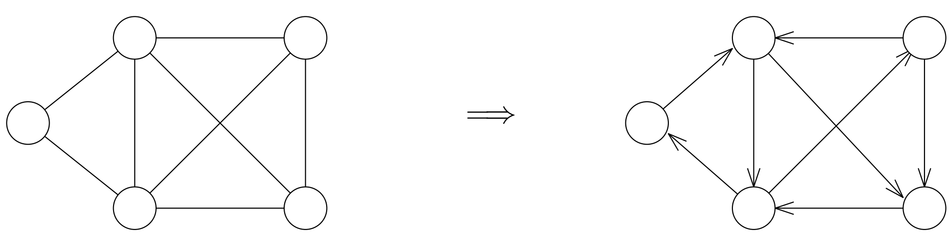
(b) (10 pts) If G is bipartite, show that the optimal total length is the same as $2k^*$, where k^* is the maximum matching size. (So, the problem is equivalent to maximum matching in the bipartite case.)

(c) (70 pts) Now consider the problem in the general case when G is not bipartite. Show that the problem can also be reduced to bipartite maximum matching and so can be solved in $O(mn)$ time by the Hungarian method or $O(m\sqrt{n})$ time by Hopcroft and Karp's algorithm.

(Hint: Construct a new bipartite graph G' where we create two versions of each vertex and two versions of each edge... Carefully prove correctness.)

Problem 7.2: Given an undirected graph $G = (V, E)$ and an integer d, we want determine whether it is possible to direct the edges so that the resulting directed graph has maximum **out-degree** at most d. Describe how to solve this problem by reduction to maximum flow. Prove correctness of your method.

Example: for the undirected graph below (left) and $d = 2$, the answer is yes, and one solution is shown on the right (there are many other solutions).



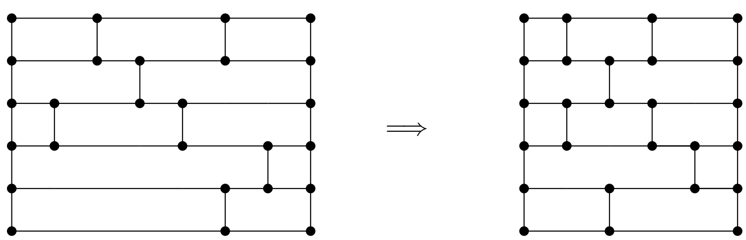
(Hint: start with a bipartite graph where the vertices on the left side are *edges* in G and the vertices on the right side are the vertices in G; add a source and a sink...)

Problem 7.3: We are given a set of n axis-aligned rectangles $R = \{r_1, \ldots, r_n\}$, where each rectangle r_i has a cost a_i. The rectangles r_i all have height 1 and have integer x- and y-coordinates, and no two rectangles intersect except along the boundaries, and their union is a rectangle $U(R)$. We want to redraw the rectangles as $R' = \{r'_1, \ldots, r'_n\}$, whose union is a new rectangle $U(R')$, so that the rectangles r'_i still have height 1 and have integer x- and y-coordinates, and the adjacency structure is unchanged, i.e., if a side of the rectangle r_i touches a side of rectangle r_j, then the corresponding side of r'_i touches the corresponding side of r'_j.

Describe a polynomial-time algorithm to compute a redrawing S' that minimizes $\sum_{i=1}^{n} a_i \cdot \mathrm{width}(r'_i)$, where $\mathrm{width}(r'_i)$ denotes the width of r'_i.

You may assume that there is a polynomial-time algorithm for the following version of the *minimum-cost flow* problem: given a directed graph $G = (V, E)$ with source s and sink t and a value d, where each edge $e \in E$ is given numbers $\ell(e), c(e), \mathrm{cost}(e)$, compute a flow f of value d, minimizing $\sum_{e \in E} \mathrm{cost}(e) \cdot f(e)$, such that for every $e \in E$, we have $\ell(e) \leq f(e) \leq c(e)$, and we have conservation of flow at every vertex in $V - \{s, t\}$.

(Hint: create a vertex for each rectangle. . .)



a redrawing with smaller width