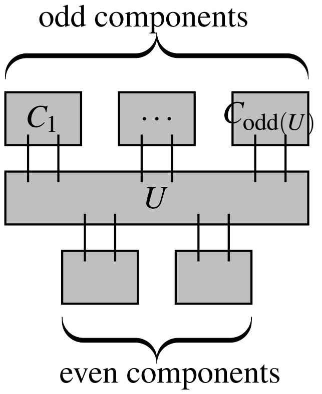
Lecturer: Thomas Rothvoss | Due date: Friday, Mar 14, 2025, on GradeScope

Problem Set 8

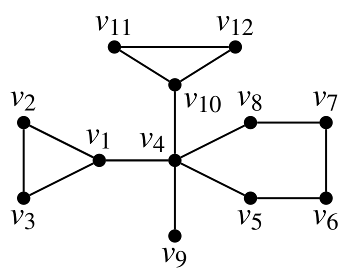
409 - Discrete Optimization

Winter 2025

Exercise 1 (10 points)
Let $G=(V,E)$ be an undirected graph. For $U \subseteq V$, let $\text{odd}(U)$ be the number of connected components in $G \setminus U$ that have an odd number of vertices. Below is a schematic picture.



a) In the example graph below, find a node set U so that $\text{odd}(U) > |U|$.

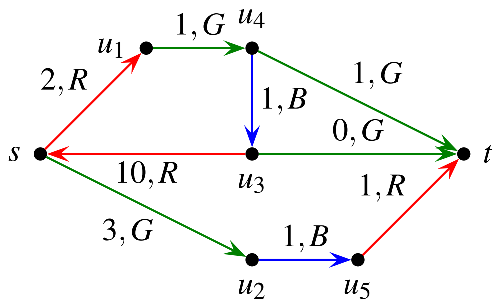


Remark: Note that together with b), this proves that this particular graph does not contain a perfect matching.

b) Now consider an arbitrary graph $G=(V,E)$. Prove that if there is a set $U \subsetneq V$ with $\text{odd}(U) > |U|$, then G does not contain a perfect matching.

Exercise 2 (10 points)
Let $G=(V,E)$ be a directed graph with non-negative edge cost $c : E \to \mathbb{R}_{\geq 0}$ and let $s,t \in V$ be two distinguished nodes. Moreover edges are labelled with one of three colors, i.e. we have a function $f : E \to \{\text{red}, \text{green}, \text{blue}\}$. A *red-green-blue walk* is a sequence of edges $e_1, \ldots, e_k \in E$ with $e_i = (v_{i-1}, v_i)$ for $i \in \{1, \ldots, k\}$ so that the colors of the visited edges are red-green-blue-red-green-blue-red-... (more formally $f(e_{3j+1}) = \text{red}, f(e_{3j+2}) = \text{green}, f(e_{3j+3}) = \text{blue}$ for integers j). The cost of such a walk is $\sum_{i=1}^{k} c(e_i)$.

a) What is the minimum cost red-green-blue walk from s to t in the depicted instance? Here we label each edge e with the cost $c(e)$ and the color $f(e)$.



Hint: The cost should be 19.

b) Design a polynomial time algorithm for finding the minimum cost red-green-blue walk from s to t in a general graph $G = (V, E)$. Explain the running time bound that you get. Argue why the algorithm is correct.
Hint: It might be worth reading again the algorithm to find M-alternating walks in Section 8 of the lecture notes.