CECS 528, Homework Assignment 6, Spring 2025, Dr. Ebert

Directions: Please review the `Homework` section on page 6 of the syllabus including a list of all rules and guidelines for writing and submitting solutions.

Due Date: Saturday April 18th as a PDF-file upload to the HW6 Canvas dropbox.

Problems

1. Consider the following optimization problem which we'll call `W`. An instance of `W` is an undirected weighted graph $G = (V, E, w)$. Moreover, for $S \subseteq V$, we define

$$w(S) = \sum w(e),$$

where the sum is taken over all edges $e = (u, v) \in E$ for which $u \in S$ and $v \notin S$. The goal is determine S so that $w(S)$ is maximized. The following is an approximation algorithm for computing the optimal set. On input G, start with any set $S \subseteq V$. Is there a set T for which either i) $T = S \cup \{v\}$ for some $v \in V - S$, or ii) $T = S - \{v\}$ for some $v \in S$ and for which $w(T) > w(S)$? If yes, then $S \leftarrow T$ and repeat. Otherwise return S.

   (a) Starting with $S = \emptyset$, demonstrate the algorithm with the graph $G = (\{1, \ldots, 9\}, E, w)$ having the weighted edges

$$E = \{(1,3,29),(1,2,28),(1,6,26),(1,9,21),(2,4,23),(2,6,25),(2,9,27),(3,4,19),(3,6,13),$$
$$(3,8,14),(4,7,11),(4,8,21),(5,6,17),(5,7,26),(5,9,27),(7,8,14),(7,9,10),(8,9,20)\}.$$

   Include a table that shows the inter and intra weight sums of each vertex before each round and add to (respectively, remove from) S the vertex v in $V - S$ (respectively, S) whose intra-weight-sum minus inter-weight-sum is a maximum. For example, v's intra-weight-sum equals the sum of all weights of edges that are incident with v and another vertex in $V - S$ (respectively, S). For example, if $S = \emptyset$, then the intra-weight-sum of vertex 1 equals $29 + 28 + 26 + 21 = 104$ while its inter-weight-sum equals 0. (10 pts)

   (b) Prove that the algorithm has an approximation ratio equual to 2. Is this algorithm guaranteed to run in polynomial time? Explain. (20 pts)

2. Consider the following optimization problem which we'll call `L`. An instance of `L` is an undirected graph $G = (V, E)$ along with three vertices $a, b, c \in V$. The problem is determine the minimum number of edges need to be removed from E so that each of a, b, and c is in a different connected component.

   (a) Describe an approximation algorithm for L whose approximation ratio is at most 2. Prove that your algorithm does indeed have the desired ratio bound. (20 pts)

   (b) Demonstrate your algorithm on the graph G (without the weights!) from part a) of Problem 1. (10 pts)