ACO Comprehensive Exam Spring 2023

Jan 5, 2023

1 Algorithms

Super Mario Secret Level.

Super Mario unlocks a new world, where he runs with Bowser on a weighted, directed graph. Each directed edge has two positive weights (m_e, b_e) corresponding to Mario and Bowser, respectively. When the arch-enemies traverse an edge, they multiply their respective number of coins by their corresponding weight on that edge. For example, if they are at vertex u and they each have ten coins, when traversing the directed edge $e = (u, v)$ with weights $(2, 1.5)$, Mario ends at vertex v with 20 coins and Bowser gets there with 15 coins. They always run together.

You are given the directed, weighted graph $\{G = (V, E), (m_e, b_e),\ e \in E\}$. Both Mario and Bowser start at vertex $s \in V$, and have one coin each to start.

(a) (2 points) Suppose Mario and Bowser take the path $\mathcal{P} = \{e_1, e_2, e_3, \dots, e_k\}$ ending at vertex $v \in V$. Give a formula for the ratio of the number of coins Bowser has at vertex v to the number of coins that Mario has at vertex v.

(b) (2 points) You are told that $b_e > 1$ for all $e \in E$ (i.e., Bowser will always increase his number of coins!). Mario can choose the path, but will be able to rescue the princess only if Bowser gets to the special vertex $t \in V$ with less than 100 coins. Design an algorithm that takes as input the weighted graph G and outputs whether or not Mario can rescue the princess.

(c) (6 points) In the bonus level, you are given a graph with positive edge weights (not necessarily bigger than one). This time, Mario will rescue the princess iff there is an *infinite* walk such that the ratio of the number of coins Mario gets to the number of coins Bowser gets tends to infinity. Design an algorithm to decide if it is possible for Mario to rescue the princess.

To earn full credit, you must describe your algorithm in detail and justify its correctness. You should also state and analyze its complexity, which should be polynomial in the size of the graph.

2 Graph Theory

For positive integers s, t, let $R(s,t)$ denote the least integer n such that any graph on n vertices contains an independent set of size s or a clique of size t. For a family of graphs $\mathcal{F}$ we say that a graph G is $\mathcal{F}$-free if no induced subgraph of G is isomorphic to a member of $\mathcal{F}$. For parts (1) and (2), let s, t be integers with $s, t \geq 3$.

(1) Let G be a $\{K_{1,s}, K_{t+1}\}$-free graph. Show that $\chi(G) \leq R(s,t) - 1$.
(Hint: first show that $\Delta(G) \leq R(s,t) - 1$.)

(2) Show that there exists a $\{K_{1,s}, K_{t+1}\}$-free graph G such that $\chi(G) \geq \frac{R(s,t+1)-1}{s-1}$.

(3) Suppose G is $\{K_{1,3}, K_4, W_5\}$-free, where W_5 denote the wheel on 6 vertices. Show $\chi(G) \leq 4$.
(W_5 is the graph containing a 5-cycle and one additional vertex connected to all vertices of the 5-cycle).

3 Linear Inequalities

Suppose we have a collection of closed, finite intervals of the real line $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$, with $I_i \subset \mathbb{R}$.

(i) (3 points) We want to find the maximum cardinality sub-collection of intervals that are disjoint, i.e., each point on the real line lines in at most one interval in the collection in which there is at most one of the chosen intervals. Write a linear program for this so that the extreme points of its feasible region correspond to subsets of disjoint intervals.

(ii) (7 points) Suppose that every point on the real line lies in at most k intervals of $\mathcal{I}$. Show that $\mathcal{I}$ can be partitioned into at most k sets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$ such that each $\mathcal{I}_k$ is a collection of non-overlapping intervals.