

CS 670 (Fall 2022) Midterm (Takehome) — Due 10/24/2022 in class

This exam is takehome, but apart from having more time and getting to work on it wherever you want, you should treat it exactly like an open-book in-class exam. Specifically, here are the key rules. (If in doubt, check with the instructor.)

- (a) You may access the textbook, anything we handed out or posted for this class (assignments, sample solutions, class notes), and anything you wrote for this class (class notes, homework solutions). You may not access any other books, any online resources (not even Wikipedia), notes taken by classmates or friends, etc.
- (b) You may not communicate about this exam with anyone except the instructor and TA. Not with classmates, labmates, friends from undergraduate, or anyone else. Not even basic questions — you should communicate exactly as if you were sitting in an in-class exam, i.e., not at all.
- (c) You may ask the TA and instructor questions in person, by e-mail or on Piazza. If you use Piazza, ensure that your post is private and only visible to the instructor and TA.
- (d) The exam is due at the beginning of class on Monday, 10/24/2022. Late submissions will not be accepted. If you are traveling, you must make arrangements with the instructor ahead of time, and may submit by e-mail, but only before the deadline.

There are 3 pages and 4 questions.

G O O D L U C K

(1) [5+5=10 points]

An interesting question about a directed graph is whether it can be covered by cycles. This question has recently gained prominence for its role in multiway swap barter economies. For instance, suppose that we have a graph whose nodes are people and their houses. (Node v describes both the person and the house.) There is a directed edge from node u to node v if u would be interested in living in the house currently inhabited by v . (There may be self-loops, if people like their own houses, but of course, the existence of self-loops is not guaranteed.) The question is now whether there is a way to reassign houses so that everyone ends up in a house they like.

Formally, we are given a directed graph $G = (V, E)$ (self loops are possible), and ask whether there is a collection of cycles C_1, C_2, \dots in G such that each node is in *exactly* one of the cycles C_j . (Notice that just a self-loop is a cycle as well.)

- (a) [5 points] Give a polynomial-time algorithm for deciding whether G has such a collection of cycles. Prove your algorithm correct. (Hint: Matchings)
- (b) [5 points] Give a characterization in the style of Hall's Theorem stating when a graph can be covered in this way. In other words, find out what the "obstacles" are for solving this problem, and characterize a directed graph as being coverable if and only if it has none of the "obstacles."

(2) [10 points]

In class, we proved that the VERTEX COVER problem is NP-complete in general. However, for some classes of graphs, it can be solved in polynomial time. This includes graphs of small (constant) bandwidth. A graph is said to have bandwidth b if the nodes can be sorted as v_1, v_2, \dots, v_n such that each edge $e = (v_i, v_j)$ has $|j - i| \leq b$. In other words, the graph can be drawn on a line so that all edges are "short."

Prove that if G is an undirected graph with vertex costs $c_v \geq 0$, and G has bandwidth at most 3, then a minimum-cost vertex cover can be found in polynomial time. You may assume that along with G , you are actually given a vertex ordering of bandwidth at most 3. (This result holds for any constant bandwidth, but you only need to show it for bandwidth 3 here.)

(3) [10 points]

The adoption of technologies in a society depends on the social structure. For example, computers, social networking sites, or programming languages derive their utility not only from their intrinsic value, but also from one's ability to collaborate or share with other people.

We can model this using a bidirected graph G in which each node is an individual. For each friendship, there are two directed edges (u, v) and (v, u) . There are two different technologies, A and B. For each directed edge $e = (u, v)$, there is a non-negative 2×2 matrix $P^{(e)}$ capturing the impact on u of the interactions between the two endpoints: for instance, $p_{A,A}^{(e)}$ captures the utility to node u if both nodes choose technology A. Similarly, $p_{A,B}^{(e)}$ is the utility to u if node u chooses A and node v chooses B.

We assume that you prefer for your friend to use the same technology as you, so $p_{A,A}^{(e)} \geq p_{A,B}^{(e)}$ and $p_{B,B}^{(e)} \geq p_{B,A}^{(e)}$. We make no assumptions about any other relationships between the matrix entries; in particular, different nodes could prefer different technologies a priori; there is no assumption of any kinds of "symmetry", etc.

We are now interested in finding an assignment $T(v) \in \{A, B\}$ of the two technologies to the nodes that maximizes the sum of total utilities of all nodes, which is $\sum_{e=(u,v)} p_{T(u),T(v)}^{(e)}$. Give and analyze a polynomial-time algorithm for finding such an assignment.

(4) [1+2+7=10 points]

A popular area of study at the intersection between psychology and computer science is understanding human preferences. Typically, for each person in isolation (keeping the context constant), preferences

are acyclic: it would be rare that someone prefers strawberries to peaches, peaches to bananas, and bananas to strawberries. Of course, different people may have very different preferences/orderings over items, but sociologists and psychologists have found that “similar” people often have similar preference orders: for instance, political liberals generally prefer prog rock to country music, while conservatives prefer country music to prog rock.

A psychology researcher wants to investigate a particular version of this: she posits that there are T types of people. Each type $t = 1, \dots, T$ is characterized by an ordering π_t on a set of n items (think fruits, or types of music). Her strong hypothesis is that every person in the world belongs to one of those T types, meaning that the person has exactly that preference order. In order to test this hypothesis, the researcher set up a website on which volunteers answered questions of the form “Which of peaches and bananas do you prefer?”. When an individual visited the website, they could answer as many or as few questions as they liked, and then left. For each of the m visitors $j = 1, \dots, m$ to the website, this resulted in a set S_j of comparisons that this visitor entered. Now, the researcher wants to assign to each visitor j a type $t(j) \in \{1, 2, \dots, T\}$, and determine an ordering π_t for each type t , such that *all* of the comparisons in S_j are consistent with the order $\pi_{t(j)}$. She has been having a lot of trouble coming up with an efficient algorithm to do so, so she has turned to you for help.

- (a) Show that her problem is equivalent to the following: You are given a set of n nodes, and m subsets S_1, \dots, S_m of directed edges. You are to partition $\{1, \dots, m\}$ into T disjoint sets R_t such that for each t , the set $\bigcup_{j \in R_t} S_j$ is an acyclic edge set.
- (b) Prove that if each $|S_j| = 1$, i.e., each visitor only answered a single question, then for every input, it is possible to partition the visits using only $T = 2$ types.
- (c) Confirm why the psychologist had difficulty with her computational task, by proving the following: even for $T = 2$ types, it is NP-complete to decide if the desired partition R_1, R_2 exists. (If you do not succeed at proving NP-hardness for $T = 2$, you can get partial credit by proving NP-hardness when T is part of the input, or for a larger constant T .)

Hint: There is a not-too-difficult reduction using NAE-3-SAT, defined below. You may use the fact that NAE-3-SAT is NP-hard without proving it. Of course, if you prefer, you can use any of the other NP-hard problems you learned about in class.

NAE-3-SAT (NOT-ALL-EQUAL 3-SAT) is defined as follows: you are given a formula that looks exactly like a standard 3-SAT formula. A variable assignment satisfies a clause iff at least one literal in the clause is true, and at least one literal in the clause is false. It satisfies the formula iff it satisfies all clauses.