Instructions

Homework should be done in groups of **one to three** people. You are free to change group members at any time throughout the quarter. Problems should be solved together, not divided up between partners. A **single representative** of your group should submit your work through Gradescope. Submissions must be received by 11:59pm on the due date, and there are no exceptions to this rule.

Homework solutions should be neatly written or typed and turned in through **Gradescope** by 11:59pm on the due date. No late homeworks will be accepted for any reason. You will be able to look at your scanned work before submitting it. Please ensure that your submission is legible (neatly written and not too faint) or your homework may not be graded.

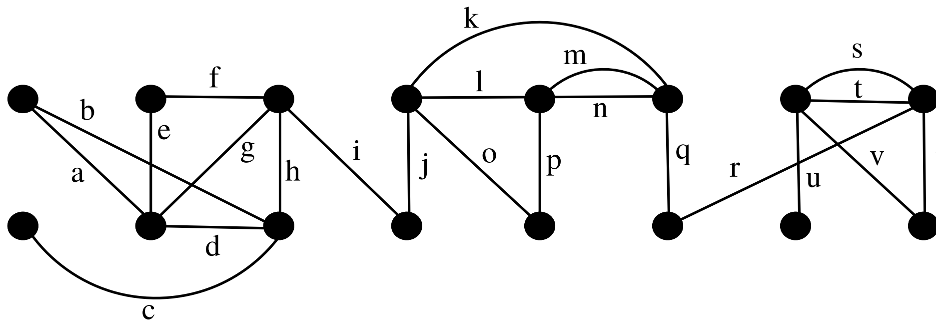
Students should consult their textbook, class notes, lecture slides, instructors, TAs, and tutors when they need help with homework. Students should not look for answers to homework problems in other texts or sources, including the internet. Only post about graded homework questions on Piazza if you suspect a typo in the assignment, or if you don't understand what the question is asking you to do. Other questions are best addressed in office hours.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

For questions that require pseudocode, you can follow the same format as the textbook, or you can write pseudocode in your own style, as long as you specify what your notation means. For example, are you using "=" to mean assignment or to check equality? You are welcome to use any algorithm from class as a subroutine in your pseudocode. For example, if you want to sort list A using InsertionSort, you can call InsertionSort(A) instead of writing out the pseudocode for InsertionSort.

Required Reading Rosen 10.1, 10.2, 10.3, 10.4 through Theorem 1, 10.5 through Example 7.

Key Concepts Graphs (definitions, modeling problems using graphs), Hamiltonian tours, Eulerian tours, Fleury's algorithm, DAGs.

1. 

   (a) (3 points) Which of the edges in the graph above are bridges?

   (b) (3 points) Use Fleury's algorithm to find an Eulerian tour of the graph above. Suppose that whenever the algorithm allows you a choice for which edge to take, you always take the edge whose label comes first alphabetically. For example, if you were in a position where you could take edge b, edge c, or edge h, you would take edge b. Write down the Eulerian tour you find by listing the edges of your tour in order.

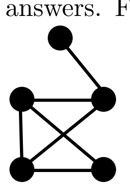
   (c) (3 points) Draw a connected graph with 5 vertices that has no Eulerian tour.

   Solutions:

   (a) c, i, j, q, r, u

   (b) c, b, a, d, h, f, e, g, i, j, k, m, l, o, p, n, q, r, s, t, w, v, u

   (c) There are several correct answers. For example, this graph has no Euler tour because four of the vertices have odd degree.



2. We say a matrix has dimensions $m \times n$ if it has m rows and n columns. If matrix A has dimensions $x \times y$ and matrix B has dimensions $z \times w$, then the product AB exists if and only if $y = z$. In the case where the product exists, AB will have dimensions $x \times w$. In this problem, we are given a list of matrices and their dimensions, and we want to determine if there is an order in which we can multiply all the matrices together, using each matrix exactly once. For example, here is a possible list of matrices and their dimensions:

   A is 3×5
   B is 4×3
   C is 4×4
   D is 2×5
   E is 5×2
   F is 5×3

   (a) (3 points) Given any list of matrices and dimensions, describe how to draw a graph so that each order in which we can multiply the matrices corresponds to a Hamiltonian tour of your graph. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

   (b) (1 point) Draw the graph described in part (a) for the example list of matrices given above.

   (c) (3 points) Given any list of matrices and dimensions, describe how to draw a graph so that each order in which we can multiply the matrices corresponds to an Eulerian tour of your graph. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

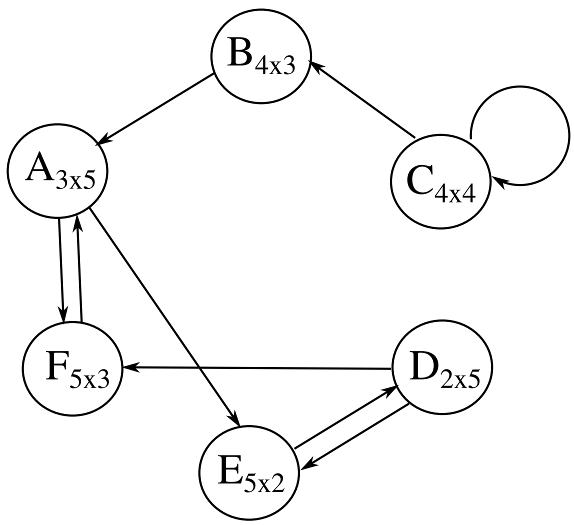
(d) (1 point) Draw the graph described in part (c) for the example list of matrices given above.

(e) (2 points) For the given example list of matrices, give one order in which we can multiply those matrices, or say that no such order exists.

Solutions:

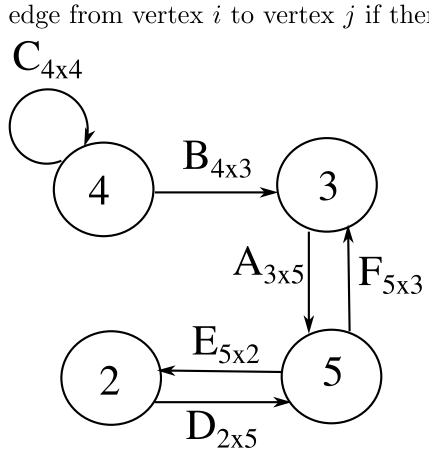
(a) Let each matrix in the list be a vertex. Draw a directed edge from matrix M to matrix N if the number of columns of matrix M equals the number of rows of matrix N, or in other words, if the matrix product MN is defined.

(b)



(c) Make a vertex for each number that appears as a dimension of a matrix in the list. Draw a directed edge from vertex i to vertex j if there is an $i \times j$ matrix in the list.

(d)



(e) $CBAEDF$ is the only order in which we can multiply the matrices. We can see this as a Hamiltonian tour in the graph from part (b), or as an Eulerian tour in the graph from part (d).

Say we have a graph where the vertices represent functions and there is an edge from f_1 to f_2 if and only if $f_1 \in O(f_2)$. When is this graph a DAG?

We claim the graph is a DAG if and only if there are no two distinct functions in the graph f_1, f_2 with $f_1 \in \Theta(f_2)$. In one direction, we show that if the graph is a DAG, there are no two such functions by proving the contrapositive: if there are two such functions, the graph is not a DAG. Assume there are two such functions $f_1 \in \Theta(f_2)$. Then since $f_1 \in O(f_2)$ and $f_2 \in O(f_1)$, there are edges from the vertex represnting f_1 to that representing f_2 and back, which is a cycle. Therefore G is not a DAG.

In the other direction, we show that if there are no such functions, then the graph is a DAG, again by proving the contra-positive: If the graph is not a DAG, then there are two functions that are Θ

of each other. If the graph is not a DAG, then there is a cycle, $v_1 \leftarrow v_2 \leftarrow v_3 ... \leftarrow v_k \leftarrow v_1$, with associated functions $f_1 ... f_k$. We prove by induction on i that $f_1 \in O(f_i)$ for $i = 1..k$. Any function is in O of itself, so the base case is true. Say that $f_1 \in O(f_i)$. Since $f_i \in O(f_{i+1})$, and O is transitive, $f_1 \in O(f_i)$. So by induction the claim holds for each i. In particular, $f_1 \in O(f_k)$, and $f_k \in O(f_1)$ since there is an edge from v_k to v_1. Thus, $f_1 \in \Theta(f_k)$, so there is such a pair of functions.

3. (10 points) Say that two actors are co-stars if they have been in the same movie. Show that in any group of six actors, we can either find a group of three such that all pairs in the group are co-stars, or a group of three so that no two in the group are co-stars.

   Solution: We want to show that in any group of six actors, at least one of these situations occurs:
   (a) There is a group of three actors such that all pairs in the group are co-stars.
   (b) There is a a group of three actors so that no two in the group are co-stars.

   Choose one particular actor, who we will call actor A. Exactly one of these two things is true:
   (i) Actor A has 3 or more co-stars among the group of 6.
   (ii) Actor A has less than 3 co-stars among the group of 6.

   In Case (i), consider the co-stars of actor A. If no two of them have been co-stars of one another, then since there are at least three of them, this forms a group of three such that no two in the group are co-stars, situation (b). If some two of them have been co-stars of one another, then those two together with actor A form a group of three such that all pairs in the group are co-stars, situation (a).

   In Case (ii), if A has less than 3 costars among the group of 6, that means there are at least three people with whom he is not a co-star. Consider these non-co-stars of actor A. If all of them have been co-stars of one another, then since there are at least three of them, this forms a group of three such that all pairs in the group are co-stars, situation (a). If some two of them are not co-stars, then those two together with actor A form a group of three in which no two in the group are co-stars, situation (b).

   Thus, in all cases, we can always find situation (a) or (b), which is what we were trying to prove.

   Note: We can formulate our answer as a problem of graph theory by constructing an undirected graph as follows. Let the vertex set V be the set of six actors. Connect two actors with an edge if they are co-stars. We must show that one of the following two situations occurs:
   (a) The graph contains a triangle, i.e. three vertices which are all connected.
   (b) The graph contains an anti-triangle, i.e. three vertices, none of which are connected.

   Choose one particular vertex, let's say v, and then the cases become:
   (i) degree(v)≥ 3
   (ii) degree(v)< 3

   The argument is exactly the same, but viewing it as a problem about graphs might help you to visualize what is going on, and it also frames the question in terms of abstract objects and relationships, not just the special case of actors with a co-star relationship that we have addressed here.

4. Prove that any tournament has a Hamiltonian path.

   We prove this by induction on n , the number of vertices in the tournament. The smallest tournament has $n = 2$, so to prove the base case, consider any tournament with two vertices, u and v. If there is an edge from u to v, (u, v) is a Hamiltonian path. Otherwise, by the definition of tournament, there is an edge from v to u, and v, u is such a path.

   Assume the claim is true whenever $n = k$, and consider a tournament on $k + 1$ vertices. Let v be a vertex, and consider the tournament on the remaining k vertices other than v. By the induction hypothesis, there is a Hamilitonian path in this sub-tournament, which passes through all the other vertices in some order, $v_1 \leftarrow v_2 \leftarrow \ldots \leftarrow v_k$.

If there are no edges from v to any v_i, then there must be an edge from v_k to v. Then $v_1 \ldots v_k, v$ is a Hamiltonian path covering all $k+1$ vertices.

Otherwise, let i be the smallest number so that there is an edge from v to v_i. If $i = 1$, there is an edge from v to v_1, and $v, v_1 \ldots v_k$ is a Hamiltonian path. Otherwise, there is an edge from v to v_i, but no edge from v to v_{i-1}. By the definition of tournament, then, there is an edge from v_{i-1} to v_i. So then $v_1 .. v_{i-1} v v_i \ldots v_k$ is a Hamiltonian path for the entire tournament.

So in all cases, there is a Hamiltonian path for the tournament of size $k+1$. Thus, by induction, there is a Hamiltonian path for any tournament of any size n.