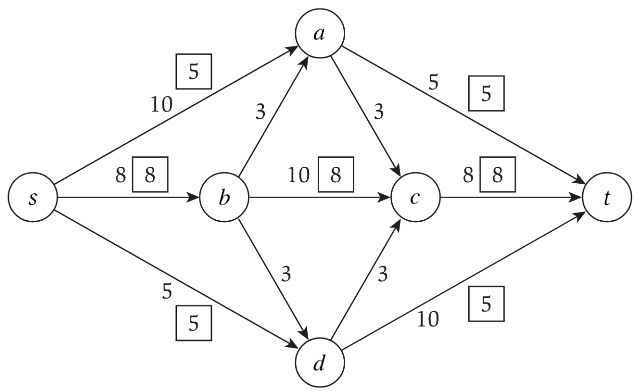
Section 10: Final Review

1. Short answer

These sample questions relate only to material after the midterm, but the real exam will be cumulative.

(a) (KT 7.2) The figure below depicts an instance of maximum flow with the original graph and capacities. The values in the squares denote the amount of flow currently being sent through each edge. Edges with no square currently have no flow being pushed through them.



◯ The flow depicted is a maximum flow.

◯ The flow depicted is not a maximum flow.

Briefly justify your answer.

(b) (KT 8.1) Recall the Interval Scheduling problem: Given a collection of intervals and an integer k, determine if the collection contains at least k nonoverlapping intervals.

(i) Does Interval Scheduling $\leq_p$ Vertex Cover?

◯ Yes

◯ No

◯ Unknown, because the answer would resolve P vs. NP

Briefly justify your answer.

(ii) Does Independent Set $\leq_p$ Interval Scheduling?

◯ Yes

◯ No

◯ Unknown, because the answer would resolve P vs. NP

Briefly justify your answer.

(c) Recall the Set Cover problem: Given a collection of sets containing objects, determine the minimum number of sets needed to cover all objects. A greedy attempt for Set Cover is:

```
1: while there exists an uncovered object do
2:     choose a set that covers the most number of still-uncovered objects
```

Suppose you are given an instance of Set Cover in which every set contains exactly 2 elements. Then this algorithm returns a set cover that is at most a factor 2 larger than the minimum set cover.

◯ True

◯ False

Briefly justify your answer.

2. Dynamic programming

A version of this problem appeared on the Section 5 handout.

Given two strings, $s = s_1 \ldots s_m$ with length m and $t = t_1 \ldots t_n$ with length n, find the length of their longest common subsequence. (A subsequence may not be contiguous. That is, one finds a subsequence by taking any subset of the indices, and putting together the letters at those indices in their original order.)

Here are a few examples:

- **Input**: $s =$ `backs`, $t =$ `arches`
  Solution: The longest common subsequence is `acs`, so the output should be 3.
- **Input**: $s =$ `skaters`, $t =$ `hated`
  Solution: The longest common subsequence is `ate`, so the output should be 3.

This problem can be solved with dynamic programming. Give a recurrence, including the base cases, that would be the basis for a dynamic programming algorithm. You should state in what order you will evaluate the subproblems, and briefly explain why your recurrence is correct.

3. Network flows

A group of traders are leaving Switzerland, and need to convert their cash in Francs (the local currency) into various international currencies. Meanwhile, the central bank is updating the security of its cash bills, and needs to collect as much cash in Francs as possible in order to efficiently retire the old bills.

There are n traders and m currencies. Trader i has T_i Francs to convert. The bank has C_j of currency j, and the exchange rate is R_j of currency j for every 1 Franc. Trader i is traveling to multiple countries and is willing to trade anywhere from L_{ij} to H_{ij} of their Francs for currency j. For example, a trader with 1000 Francs might be willing to convert between 300 and 700 of their Francs for US dollars, between 200 and 500 of their Francs for Euros, and exactly 0 Francs for Japanese yen. It would be valid to have them trade 400 Francs for US dollars and 400 Francs for Euros. Assume that $\sum_j L_{ij} \leq T_i$.

All traders give their requests to the bank at the same time, and the bank is deciding how to fulfill them. Describe an efficient algorithm that determines whether or not the bank can satisfy all requests, and if so, a method of satisfying the requests to maximize the amount of Francs it collects. Briefly explain all the choices made in your algorithm.

4. Linear programming

A version of this problem appeared on the Section 8 handout.

You are a politician running for local office, and you want to appeal to a wide voter base. There are k groups of voters, let m_i be the number of voters in the ith group, and you want at least half of each group to vote for you. Without any campaigning, a_i voters from group i will vote for you ($0 \leq a_i \leq m_i$).

Your campaign staff have determined that there are n issues that voters care about, and they will react differently depending on their group. In particular, for every \$1000 you spend on advertising for issue j, d_{ij} is the number of additional voters in group i who will now vote for you. (If d_{ij} is negative, it means you lost voters in group i.)

Write a linear program in standard form to determine the minimum advertising cost so that at least half of each group votes for you, if possible at all. Briefly explain all choices made in your program.

5. Reduction

Consider the following problems:

HAMILTONIANPATH
Input: A directed graph G
Output: Determine if there is a Hamiltonian path in G (a path that visits each vertex exactly once).

HAMILTONIANCYCLE
Input: A directed graph G
Output: Determine if there is a Hamiltonian cycle in G (a cycle that visits each vertex exactly once).

Suppose that HAMILTONIANPATH is NP-hard. Use that fact to show HAMILTONIANCYCLE is NP-hard.

6. Bonus: Dynamic programming, again

The inclusion of this problem is simply because the problem and solution are existing from previous iterations of this course, and should not be interpreted as indicating stronger emphasis on dynamic programming on the final exam.

The problem is to determine, in a fictional country with two-party elections and an electoral college, the smallest number of votes needed to win the election. Let the number of states in this country be n.

Let p_i be the total number of voters participating in the election from state i, and let v_i be the number of electoral votes for state i. Each state holds a statewide election between the two candidates, and assume that there are no state-level ties, so candidates must win $\lfloor p_i/2 \rfloor + 1$ votes in state i to win it. All electoral votes of a state go to the candidate winning that state. If a candidate receives at least V electoral votes in total, where $V = \lfloor (\sum_i v_i)/2 \rfloor + 1$, they win the overall election.

Determine the minimum percent of the total popular vote that a candidate must obtain in order to receive at least V electoral votes and win the overall election.