This closed-book, closed-notes two-hour test consists of 6 questions, the last of which is for extra credit. The number of points for each problem is indicated on the next page.

- Read all questions carefully before starting.
- Work on the problems that seem easiest first.
- Attempt to solve all problems.
- Show your work, but also remember that concise answers are preferable to wordy ones.
- Write all your answers clearly on the space provided in the exam paper. If you need additional paper, please ask us.
- If you do not understand a problem, please ask us for clarification.
- Clearly state any simplifying assumptions you make in solving a problem.
- When asked to describe algorithm, you are expected to argue its correctness and analyze its running time.
- When doing reductions, you may use any of the problems discussed in the text or in the homework. These problems include Vertex Cover, Independent Set, Clique, Hamiltonian Cycle, 3-SAT, Circuit SAT, Set Cover, Hitting Set, Subset Sum, Number Partitioning, 3-Coloring, and 3-Dimensional Matching.
Score

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

Notes

- Question 6 is for extra credit.
- Extra credit will be added to your total score, for a maximum of 120 points out of 100.
1 Max 3-SAT (20 points)

In the Max 3-SAT problem we are given a 3-CNF formula $\varphi(x_1, x_2, \ldots, x_n)$ with $m$ clauses and an integer $K$. The goal is to determine if there exists a truth assignment that makes at least $K$ of the clauses evaluate to true.

(a) (6 points) Show that Max 3-SAT is in NP.

(b) (14 points) Prove that Max 3-SAT is NP-complete by providing an appropriate reduction.
2 Choosing a construction crew (20 points)

The ACME construction company is choosing a crew for a large and potentially lucrative project from a set $P$ of $n$ workers. To help in making its decision, the company has an $n \times n$ table, called a compatibility matrix that, for every pair of people, has a 1 if the two are compatible and a 0 otherwise. A consultant argues that, since finding the largest subset of compatible workers seems extremely difficult, the company should look for the smallest subset $D \subseteq P$ whose removal from $P$ leaves a compatible set. More formally, the problem is as follows: Given a compatibility matrix for a set $P$ of $n$ workers and an integer $k$, does there exist a subset $D \subseteq P$ of size at most $k$ such that every pair of workers in $P - D$ is compatible? For lack of a better term, let’s call this the consultant’s problem and abbreviate it CP.

(a) (6 points) Show that CP $\in$ NP

(b) (14 points) Show that CP is NP-complete.
3 Two-processor load balancing (20 points)

Suppose we are given a set of \( n \) jobs, where job \( i \) has processing time \( t_i \). The goal is to assign each job to one of the two machines so as to minimize the maximum load on a machine. Formally, for \( j = 1, 2 \), let \( A(j) \) be the set of jobs assigned to machine \( j \), and define the load on machine \( j \) as

\[
T_j = \sum_{i \in A(j)} t_i.
\]

The makespan of the schedule is the maximum load of a processor, that is,

\[
T = \max(T_1, T_2).
\]

The problem is to find an assignment of jobs to processors that has minimum makespan.

(a) (4 points) Formulate a decision version of the two-processor load balancing problem.

(b) (4 points) Show that the decision version of the load balancing problem is in NP.
(c) (12 points) Show that the decision version of the load balancing problem is NP-complete.
4 Search versus decision (20 points)

Given a directed graph $G = (V, E)$, the directed Hamiltonian cycle problem (DHC) asks whether $G$ has a simple cycle that visits every node of $G$.

Suppose that you have a fast algorithm $A$ to solve DHC. This algorithm takes as input a directed graph $G$ and returns “yes” or “no” depending on whether or not the required cycle exists. Using $A$ as a black box, design an algorithm that takes as input a directed graph $G$, and does one of the following two things:

- Returns a Hamiltonian cycle for $G$, or
- Gives a (correct) message that no such cycle exists in $G$.

Your algorithm should use at most a polynomial number of steps, together with a polynomial number of calls to algorithm $A$. Remember to give an explicit correctness argument and a justification of the running time.
5  Duality (20 points)

Consider the following 1-variable linear program, which we call $P$:

\[
\begin{align*}
\text{maximize} & \quad tx \\
\text{subject to} & \quad rx \leq s \\
& \quad x \geq 0,
\end{align*}
\]

where $r \neq 0$, $s \neq 0$, and $t \neq 0$ are real numbers. Let $D$ be the dual of $P$.

(a) (4 points) Write the dual, $D$, of $P$

State (briefly justifying your answers) for which non-zero values of $r$, $s$, and $t$ you can assert that

(b) (4 points) Both $P$ and $D$ have optimal solutions with finite objective values.
(c) (4 points) \( P \) is feasible, but \( D \) is infeasible.

(d) (4 points) \( D \) is feasible, but \( P \) is infeasible.
(e) (4 points) Neither $P$ nor $D$ is feasible.
6 Duality (20 extra credit points)

In the single-pair shortest-path problem we are given a directed graph \( G = (V, E) \) with a source vertex \( s \), a destination vertex \( t \), and where each edge \( e \) has a nonnegative real-valued length \( w(e) \). We wish to compute the length of a shortest path from \( s \) to \( t \), where the length of a path is the sum of the lengths of its edges. It can be shown that this problem can be expressed as a linear program of the following form.

\[
\begin{align*}
\text{maximize} & \quad d[t] - d[s] \\
\text{subject to} & \quad d[v] \leq d[u] + w(u, v) \text{ for every edge } (u, v) \in E.
\end{align*}
\] (2)

The optimal value of the objective function can be shown to equal the length of the shortest path from \( s \) to \( t \). Note that (2) is a maximization problem; the reason for this will become apparent if you solve part (b).

(a) (12 points) Write the dual of linear program (2).

(b) (8 extra credit points) Give an intuitive interpretation of the dual.