Final Exam

Instructions. This is a three-hour exam. There are ten questions worth a total of 100 points. The last problem also includes a 5-point bonus question.

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

Name: ____________________________
1. (10 points) Prove that

\[ \sum_{1 \leq i \leq n} i^{10} = \Theta(n^{11}). \]
2. (10 points) Use a potential function argument to prove that the following program terminates from any initial program state. Remark: All variables are of type integer.

```c
i := 1;
while (i > 0) {
    if (n > 100) {
        n := n - 10;
        i := i - 1
    } else {
        n := n + 11;
        i := i + 1
    }
}
```

Hint: Make use of the potential function $2|n - 111| + 21i$. 
3. (10 points total) Let $A$ be an algorithm for some graph problem $P$, and assume that $A$ has the following characteristics:

- On any instances $I$ of $P$ involving a graph with $n \leq 100$ vertices, $A$ solves $I$ in at most $n!$ steps.
- On any instance $I$ of $P$ involving a graph with $n > 100$ vertices, $A$ solves $I$ recursively, as follows:
  1. First, algorithm $A$ uses a total of at most $10n$ steps to determine three instances $I_1$, $I_2$, and $I_3$ of $P$, each of which involves a graph with at most $\lfloor n/2 \rfloor$ vertices.
  2. Then, algorithm $A$ recursively solves instances $I_1$, $I_2$, and $I_3$.
  3. Finally, algorithm $A$ uses at most $n^2$ additional steps to determine a solution to instance $I$ from the solutions to instances $I_1$, $I_2$, and $I_3$.

Let $T(n)$ denote the worst-case running time of algorithm $A$ on any input graph with $n$ vertices.

(a) (5 points) Formulate a recurrence for upper-bounding $T(n)$.

(b) (5 points) Solve the recurrence you obtained in part (a) to obtain closed-form $O$-bound on $T(n)$.  

4. **(10 points total)** Let each of \(m\) balls be thrown independently and uniformly at random into one of \(n\) bins. For the purposes of this question, let us define a *couple* as an unordered subset of the balls of cardinality 2. We say that a couple \(C\) *collides* if the two balls belonging to \(C\) land in the same bin.

(a) (3 points) For any given couple \(C\), what is the probability that \(C\) collides?

(b) (3 points) How many couples are there?

(c) (4 points) Let the random variable \(X\) denote the total number of couples that collide. Use the results of parts (a) and (b) to compute \(E(X)\). Hint: Use linearity of expectation.
5. (10 points total) For each integer \( i \), let \( f_i(n) \) denote the function \( n^i \). Let \( A \) denote the set of functions \( \{ f_i(n) \mid i \in \mathbb{Z} \} \). For any pair of functions \( g(n) \) and \( h(n) \) in \( A \), define \( g(n) \preceq h(n) \) if and only if \( g(n) = \Theta(h(n)) \).

(a) (6 points) Is \( (A, \preceq) \) a poset? Justify your answer.

(b) (4 points) How would your answer to part (a) change if \( f_i(n) \) were defined as \( i \cdot n \) instead of \( n^i \)?
6. (10 points total)

(a) (3 points) Define the term well-ordered set.

(b) (4 points) State the principle of well-ordered induction (i.e., the induction principle for well-ordered sets).

(c) (3 points) Prove that the poset \((\mathbb{N} \times \mathbb{N}, \preceq)\) is well-ordered, where \((a, b) \preceq (c, d)\) if (1) \(a < c\) or (2) \(a = c\) and \(b \leq d\). Note: You may assume that \((\mathbb{N} \times \mathbb{N}, \preceq)\) is a poset.
7. (10 points total)

(a) (3 points) Give pseudocode for Warshall’s transitive closure algorithm. Hint: Warshall’s algorithm involves a triply-nested loop.

(b) (3 points) Suppose that in an entire iteration of the outermost loop of Warshall’s algorithm, no boolean variable has its value changed. Can we stop the algorithm at this point, i.e., is the remaining computation guaranteed to leave all of the boolean variables unchanged? Justify your answer.

(c) (4 points) Describe how to modify Warshall’s transitive closure algorithm to compute all-pairs shortest paths in a directed graph with nonnegative edge weights. Remark: Every vertex has a path of length 0 to itself (the empty path).
8. (10 points) Let $S$ be a set of $n$ points. For every pair of points $x$ and $y$ in $S$, let $d(x, y)$ denote a given real distance from $x$ to $y$. The pair $(S, d)$ is said to be a metric space if the following properties hold for all points $x, y,$ and $z$ in $S$: (1) $d(x, y) \geq 0$; (2) $d(x, y) = d(y, x)$; (3) $d(x, y) \leq d(x, z) + d(z, y)$; (4) $d(x, y) = 0$ if and only if $x = y$. Let $G = (V, E)$ be a connected undirected graph with positive real edge weights. (Note: Zero is not a positive number.) For any pair of vertices $u$ and $v$ in $V$, let $d_G(u, v)$ denote the length (i.e., total weight of) a shortest path from $u$ to $v$. Prove that $(V, d_G)$ is a metric space.
9. **(10 points total)** Recall that Dijkstra’s single-source shortest paths algorithm is designed to process directed graphs with nonnegative edge weights. In this question we investigate the behavior of Dijkstra’s algorithm when it is executed on a directed graph $G = (V, E)$ with arbitrary real edge weights. Let $s$ denote the source vertex.

(a) **(2 points)** Argue that if $G$ does not contain a negative-weight directed cycle, then for any vertex $v$ there is a simple shortest path from $s$ to $v$ (i.e., a shortest path that does not revisit any vertices).

(b) **(2 points)** Argue that if $G$ contains a negative-weight directed cycle that is reachable via a directed path from $s$, then for some vertex $v$ the shortest path distance from $s$ to $v$ is $-\infty$.

(c) **(2 points)** Argue that if all edge weights associated with $G$ are finite then for all vertices $v$, Dijkstra’s algorithm computes a finite shortest path distance from $s$ to $v$.

(d) **(2 points)** Use the result of the previous two parts to argue that Dijkstra’s algorithm does not correctly compute shortest path distances if $G$ contains a negative-weight cycle that is reachable via a directed path from $s$.

(e) **(2 points)** Give an example showing that even if $G$ does not contain a negative-weight cycle, the presence of negative-weight edges can cause Dijkstra’s algorithm to fail.
10. **(10 points total, plus a 5 point bonus)** In each of the following parts, let $G = (V, E)$ be a given directed graph.

(a) **(3 points)** Give the pseudocode for performing a depth-first search of $G$.

(b) **(3 points)** Define a relation $R$ over $V$ as the set of all ordered pairs of vertices $(u, v)$ such that there is a (possibly empty) directed path from $u$ to $v$ and a (possibly empty) directed path from $v$ to $u$. Prove that $R$ is an equivalence relation. Remark: The equivalence classes of $R$ are the vertex sets of the strongly connected components of $G$.

(c) **(4 points)** How can the depth-first search routine of part (a) be used to determine the strongly connected components of $G$ in $O(|V| + |E|)$ time?

(d) **(5 point bonus)** We say that $G$ is *semiconnected* if for all pairs of vertices $u$ and $v$, either there is a path from $u$ to $v$ or there is a path from $v$ to $u$. Give an $O(|V| + |E|)$-time algorithm for determining whether $G$ is semiconnected. Hint: Make use of part (c) to determine the strongly connected components of $G$. Then shrink these SCCs to "supervertices" to convert $G$ into a DAG $G'$. Then use depth-first search to perform a topological sort of $G'$.