Neatly print your name (first name first, with no comma), your network ID, and a short alias into the boxes above. Do not sign your name. Do not write your Social Security number. Staple this sheet of paper to the top of your homework.

Grades will be listed on the course web site by alias give us, so your alias should not resemble your name or your Net ID. If you don't give yourself an alias, we'll give you one that you won't like.

This homework tests your familiarity with the prerequisite material from CS 173, CS 225, and CS 273—many of these problems have appeared on homeworks or exams in those classes—primarily to help you identify gaps in your knowledge. You are responsible for filling those gaps on your own. Parberry and Chapters 1–6 of CLR should be sufficient review, but you may want to consult other texts as well.

Before you do anything else, read the Homework Instructions and FAQ on the CS 473g course web page (http://www.cs.uiuc.edu/class/sp06/cs473g/hw/faq.html), and then check the box below. This web page gives instructions on how to write and submit homeworks—staple your solutions together in order, write your name and netID on every page, don't turn in source code, analyze everything, use good English and good logic, and so forth.

☐ I have read the CS 473g Homework Instructions and FAQ.
"Be that as it may, it is tonight school that I owe what education I possess; I am the first to own that it doesn’t amount to much, though there is something rather grandiose about the gaps in it." – The tin drum, Gunter Grass

Required Problems

1. **A trip through the graph.**
   
   [20 Points]

   A tournament is a directed graph with exactly one edge between every pair of vertices. (Think of the nodes as players in a round-robin tournament, where each edge points from the winner to the loser.) A Hamiltonian path is a sequence of directed edges, joined end to end, that visits every vertex exactly once. Prove that every tournament contains at least one Hamiltonian path.

   ![A six-vertex tournament containing the Hamiltonian path 6 → 4 → 5 → 2 → 3 → 1.](image)

2. **Recurrences**
   
   [20 Points]

   Solve the following recurrences. State tight asymptotic bounds for each function in the form \( \Theta(f(n)) \) for some recognizable function \( f(n) \). You do not need to turn in proofs (in fact, please *don’t* turn in proofs), but you should do them anyway just for practice. Assume reasonable but nontrivial base cases if none are supplied. More exact solutions are better.

   (a) [2 Points] \( A(n) = A(\sqrt{n}/3 + \lfloor \log n \rfloor) + n \)

   (b) [2 Points] \( B(n) = \min_{0 < k < n} (3 + B(k) + B(n - k)) \).

   (c) [2 Points] \( C(n) = 3C(\lceil n/2 \rceil - 5) + n/\log n \)

   (d) [2 Points] \( D(n) = \frac{n}{n-3} D(n-1) + 1 \)

   (e) [2 Points] \( E(n) = E(\lceil 3n/4 \rceil) + \sqrt{n} \)

   (f) [2 Points] \( F(n) = F(\lfloor \log n \rfloor) + \log n \)

   (g) [2 Points] \( G(n) = n + \lfloor \sqrt{n} \rfloor \cdot G(\lfloor \sqrt{n} \rfloor) \)
(h) [2 Points] \( H(n) = \log(H(n - 1)) + 1 \)

(i) [2 Points] \( I(n) = 5I(\lceil \sqrt{n} \rceil) + 1 \)

(j) [2 Points] \( J(n) = 3J(n/4) + 1 \)

3. **Sorting functions**

[20 Points]

Sort the following 25 functions from asymptotically smallest to asymptotically largest, indicating ties if there are any. You do not need to turn in proofs (in fact, please don't turn in proofs), but you should do them anyway just for practice.

\[
\begin{align*}
\cos n + 2 & \quad \lg^* n / 2 \quad \lg n / 2 \quad \lg^* (n/8) \quad 1 + \lg \lg \lg n \\
\cos n + 2 & \quad \lg (\lg n) / 2 \quad (\lg n)! \quad (\lg^* n)^{\lg n} \quad n^5 \\
\lg^* 2^{2^{2n}} & \quad 2^{\lg n} \quad \sqrt{n} \quad \sum_{i=1}^{n} i \quad \sum_{i=1}^{n} i^2 \\
n^{7/(2n)} & \quad n^{3/(2\lg n)} \quad 12 + [\lg \lg(n)] \quad (\lg(2 + n))^{\lg n} \quad (1 + \frac{1}{15n})^{15n}
\end{align*}
\]

To simplify notation, write \( f(n) \ll g(n) \) to mean \( f(n) = o(g(n)) \) and \( f(n) \equiv g(n) \) to mean \( f(n) = \Theta(g(n)) \). For example, the functions \( n^2 \), \( n \), \( n^3 \) could be sorted either as \( n \ll n^2 \equiv \left(\frac{n}{2}\right) \ll n^3 \) or as \( n \ll n^2 \equiv n^2 \ll n^3 \). [Hint: When considering two functions \( f(\cdot) \) and \( g(\cdot) \) it is sometime useful to consider the functions \( \ln f(\cdot) \) and \( \ln g(\cdot) \).]

4. [20 Points] There are \( n \) balls (numbered from 1 to \( n \)) and \( n \) boxes (numbered from 1 to \( n \)). We put each ball in a randomly selected box.

(a) [4 Points] A box may contain more than one ball. Suppose \( X \) is the number on the box that has the smallest number among all nonempty boxes. What is the expectation of \( X \)?

(b) [4 Points] What is the expected number of bins that have exactly one ball in them? (Hint: Compute the probability of a specific bin to contain exactly one ball and then use some properties of expectation.)

(c) [8 Points] We put the balls into the boxes in such a way that there is exactly one ball in each box. If the number written on a ball is the same as the number written on the box containing the ball, we say there is a match. What is the expected number of matches?

(d) [4 Points] What is the probability that there are exactly \( k \) matches? (\( 1 \leq k < n \))

[Hint: If you have to appeal to "intuition" or "common sense", your answers are probably wrong!]
5. Graphs! Graphs!

[20 Points]

A coloring of a graph $G$ by $\alpha$ colors is an assignment to each vertex of $G$ a color which is an integer between 1 and $\alpha$, such that no two vertices that are connected by an edge have the same color.

(a) [5 Points] Prove or disprove that if in a graph $G$ the maximum degree is $k$, then the vertices of the graph can be colored using $k + 1$ colors.

(b) [5 Points] Provide an efficient coloring algorithm for a graph $G$ with $n$ vertices and $m$ edges that uses at most $k + 1$ colors, where $k$ is the maximum degree in $G$. What is the running time of your algorithm, if the graph is provided using adjacency lists. What is the running time of your algorithm if the graph is given with an adjacency matrix. (Note, that your algorithm should be as fast as possible.)

(c) [5 Points] A directed graph $G = (V, E)$ is a neat graph if there exist an ordering of the vertices of the graph $V(G) = \langle v_1, v_2, \ldots, v_n \rangle$ such that if the edge $(v_i, v_j)$ is in $E(G)$ then $i < j$.

Prove (by induction) that a DAG (i.e., directed acyclic graph) is a neat graph.

(d) [5 Points] A cut $(S, T)$ in a directed graph $G = (V, E)$ is a partition of $V$ into two disjoint sets $S$ and $T$. A cut is mixed if there exists $s, s' \in S$ and $t, t' \in T$ such that $(s, t) \in E$ and $(t', s') \in E$. Prove that if all the non-trivial cuts (i.e., neither $S$ nor $T$ are empty) are mixed then the graph is not a neat graph.