This exam contains

- this cover (page 1),
- six pages of problems (pages 2-7), and
- one blank page (after page 7).

You have 150 minutes. No notes, calculator, computer, etc. are allowed.

Feel free to ask clarification questions during the exam. If you feel that there is a mistake in how a question is posed, then certainly ask for clarification.

Except where otherwise noted, always show your work or otherwise explain your answer. A correct answer with no supporting argument may not earn much credit.

Throughout this exam, a *simple* graph is one that is undirected with no self-loops or parallel edges.

Good luck.

1. Recall that $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ for suitable n and k.

A. Prove that the equation is true, from either the definition of $\binom{n}{k}$ or the usual factorial expression.

B. Write a recursive function `choose(n, k)` that returns $\binom{n}{k}$ for all suitable values of n and k. Use the language of your choice, as long as your algorithm is utterly clear.

2. In this problem, you may assume without proof that $(3/2)^{11} \approx 86.5$ and $(3/2)^{12} \approx 129.75$. Prove that $F_n \geq (3/2)^n$ for all $n \geq 11$. Be careful, especially in your base case(s).

3. You select a d-digit number n at random (where d is large, such as 100 or 1000). Then you select $a \in \{2, \ldots, n-2\}$ at random, and find that n passes the Miller-Rabin test for the base a. Estimate the probability that n is prime, in terms of d.

4. In each part below, answer TRUE, FALSE, or PUNT. PUNT earns half credit, the correct answer earns full credit, and the incorrect answer earns no credit. No explanation is necessary.

A. In a simple bipartite graph, where the two vertex sets have sizes n_1 and n_2, the maximum possible number of edges is $n_1 n_2$.

B. If every vertex in a simple graph G has even degree, then there exists a path through the graph that traverses each edge exactly once and returns to where it started.

C. Computing the nth Fibonacci number recursively, based on $F_n = F_{n-1} + F_{n-2}$, requires time exponential in n.

D. For all $f, g : \mathbb{N} \to \mathbb{N}$, $f = \mathcal{O}(g) \Leftrightarrow g = \Omega(f)$.

E. In the integers, $\forall (m > 0)\ \forall a\ \forall b\ \forall c$, $ac \equiv_m bc \Rightarrow a \equiv_m b$.

F. In the integers, $\forall (m > 0)\ \forall a$, $a^{\phi(m)} \equiv_m 1$.

G. There exists a time-$\mathcal{O}(n^2)$ algorithm to determines whether any given simple graph G is connected, where n is the number of vertices.

H. There exists a time-$\mathcal{O}(n)$ algorithm to determines whether any given simple graph G is connected, where n is the number of vertices.

5. Draw a weighted simple graph G and specify a starting node s such that, when Dijkstra's algorithm is executed, there is some node whose distance record in $frontier$ gets changed twice after it is first set.

6. Explain how being able to factor large integers quickly would let you break RSA.

7. Give the asymptotic running times of the following algorithms. No explanation is necessary, but be sure to employ the best suitable notation $\mathcal{O}$, Ω, or Θ.

A. breadth-first search on a connected undirected graph with n vertices and m edges (in terms of n, m)

B. the `choose(n, k)` algorithm from Problem 1 (in terms of n only!)

C. the fastest multiplication algorithm for two $n \times n$ matrices that we have studied in this course (in terms of n, regarding addition and multiplication of numbers as elementary operations)

D. multiplication of two n-bit integers, by the standard algorithm (in terms of n)

E. the Euclidean algorithm (in terms of the arguments a and b, regarding division with remainder as an elementary operation)

F. the fastest algorithm for computing a^n that we have discussed in this course (in terms of n, regarding addition and multiplication of numbers as elementary operations)

G. an algorithm whose running time $T(n)$ on input of size n satisfies $T(1) = c$ and $T(n) = 3T(n/3) + cn^2$