CS344: FInal
Tuesday, Dec 21, 12.10 – 2:30 (140 min)

Instructions. Be precise and brief.

1. Rapid Questions. 5 points each.
   (a) $f(n) = n$. $g(n) = \frac{2^{\log n}}{3}$. Is $f = o(g(n))$? T/F, provide reasons.
   (b) There exists a polynomial time algorithm to determine whether an undirected graph contains a clique of size 3. T/F, provide reasons.
   (c) A clique in an undirected graph is a vertex cover of the graph's complement. Prove it.
   (d) If the Depth First Search (DFS) finishing time $f[u] > f[v]$ for two vertices u and v in a directed graph G, and u and v are in the same DFS tree in the DFS forest, then u is an ancestor of v in the DFS tree. T/F, provide reasons.

2. `Dynamic Programming. 20 Pts.` You are given an $n - by - n$ grid, where each square (i, j) contains $c(i, j)$ gold coins. Assume that $c(i, j) \geq 0$ for all squares.You must start in the upper-left corner and end in the lower-right corner, and at each step you can only travel one square down or right. When you visit any square, including your starting or ending square, you may collect all of the coins on that square. Give an algorithm to find the maximum number of coins you can collect if you follow the optimal path.

3. `Strings. 20 Pts` Consider pattern p that contains a number of *wildcard symbols* ϕ, each of which matches an arbitrary single character.

   For example, the pattern $a\phi a$ matches in the text $abaaa$ in both the 1st and 3rd positions, and the pattern $a\phi a\phi a$ matches the text $xxabacayyaaaaa$ in two positions.

   Given a text t and pattern p (with wildcards), design an algorithm to find all places t where p appears in time $O(m + qn)$ where m is the length of p, n is the length of t, and q is the number of ϕ's in p.

   Use Karp-Rabin fingerprints.

4. `Clique. 20 Pts` The yes/no clique problem is: given an undirected graph $G = (V, E)$ and a target integer k, is there a clique of size k? A clique is a set of vertices C in V such that each pair of vertices $(u, v) \in C$, is also an edge in E (thus every pair of vertices in a clique are connected by an edge).

   Show that you can find a clique of size k (when one exists) with a polynomial number of calls to the yes/no clique routine, plus polynomial additional work.