CS180 — Algorithms and Complexity
HW#1 — Graph Algorithms
Due: 11:55pm Monday February 2, 2015

D. Stott Parker, Yuh-Jie Chen, Xiaoran Xu
stott@cs.ucla.edu, eyjchen@cs.ucla.edu, xrxu@cs.ucla.edu

Using CCLE, please upload your answers as a PDF document before the deadline.

Please make your answers readable. Your answers should be short — in many problems a single number is enough; no long explanations are needed.

Programming is not necessary. The airport data files contain enough information for answering these questions.

1. **Recurrences**

   (a) What is the next number in each of the following sequences?

      i. 1, 11, 21, 1211, 111221, 312211, ...
      ii. 10, 9, 60, 90, 70, 66, ...
      iii. 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, ...
      iv. 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, ...

   Your answer can simply be a number; a lengthy explanation is not needed.

   (Hint: The first two are famous interview questions. The Google aptitude test might help. The last two are famous sequences.)

   (b) Give an explicit solution (nonrecursive expression) for $T(n)$, assuming that n is a power of d:
   $T(n) \ = \ k\,T(n/d) \ + \ f(n), \quad T(1) \ = \ c.$

   (c) For each of the following recurrences, derive a solution $T(n) = \Theta(...)$ using the Master Theorem, and determine which has lowest asymptotic complexity.

      i. $T(n) \ = \ 8\,T(n/2) \ + \ 4\,n^2, \quad T(1) \ = \ 1$
      ii. $T(n) \ = \ 7\,T(n/2) \ + \ 18\,n^2, \quad T(1) \ = \ 1$
      iii. $T(n) \ = \ 143640\,T(n/70) \ + \ O(n^2), \quad T(1) \ = \ 1$

   (d) Give an explicit formula involving binomial coefficients for $T\,(n) \ = \ \sum_{k=1}^{n-1} T(k)\,T(n-k), \ \ T(1) \ = \ 1.$
   (Hint: generate initial values of $T(n)$ and use oeis.org.)

2. **iPhone Stress Testing (interview question)**

   You are working at a company in a skyscraper with n floors, and the management wants to know how well iPhones do if they are thrown out of very high windows. The idea is that the iPhones aren't affected at all by this below some specific floor F. When dropped from any floor below F, they do not break; when thrown from floor F and above, they break.

   Your problem is to discover the value of F at which iPhones break. Clearly, you want to determine the value of F with as few iPhones as you can. If an iPhone doesn't break, you can use it in another trial, but of course once it breaks you can't use it again.

   More specifically, in a building with n floors, you are given k iPhones and need to determine the value of F.

   If $k = 1$, one strategy is to try dropping the iPhone from floor 1, then from floor 2, then from floor 3, ..., until it breaks. This permits you to determine the value F with 1 iPhone, but it takes F trials.

   If $k = 2$ and there are 100 floors in the building, you could use a faster strategy: use one iPhone to find the first digit of F (i.e., the 'tens digit' of the floor number), and then use the other iPhone to find the second digit.

   In a building with n floors, then, answer these questions:

   (a) if you have $k = 1$ iPhones, explain why you must use $\Omega(n)$ trials to determine F in the worst case.
   (b) if you have $k = 2$ iPhones, develop a strategy that uses $O(\sqrt{n})$ trials to determine F.
   (c) if you have $k \geq 3$ iPhones, develop a strategy that uses $O(k\,n^{1/k})$ trials.

   (Hint: for $k = 2$, the two-digit approach above might lead to the right idea, but it relies on the fact that there are $n = 100$ stories. Here we wants a solution that works in any building, and for any value of n.)

3. **Air Travel Networks**
You are given the graph of world airports and flight routes, specified in two files: `airport_connections.csv` and `airports.csv`. The first specifies nodes of the graph, and the other specifying edges between — real connections flown by existing airlines. (The data was obtained from openflights.org/data.html.)

Altogether there are 321 airports and 8542 connections (edges). However, these edges include both directions, so there are 8542/2 = 4721 connected pairs of nodes, viewing it as an undirected graph.

In the attached files, there are also files like distance_matrix.csv, giving the shortest-path distances (in hops) between any two nodes, and biconnected_components.csv, giving information about the graph. These files should eliminate the need to do programming for this problem.

(a) Find the length of the route with the fewest hops from Dakar (DKR) to Los Angeles (LAX) using established flight connections. State the distance in hops.

(b) Finding a route with the *most* hops from DKR to LAX, visiting airports at most once, is an instance of the *Longest Path problem*. What is the time complexity of the Longest Path problem?

(c) The *diameter* diam(G) of a graph G is the maximum of all pairwise distances (in hops) between nodes in any connected component; the *average pairwise distance* apd(G) is the average of these distances. What are the diameter and apd of the airport graph? Question 3.8 in [KT] asks whether the ratio diam(G)/apd(G) can be upper-bounded by a constant. Prove either there is such a bound, or that there is no such bound.

(d) The *degree distribution* of a graph is a histogram of node degrees for nodes in the graph (airport_node_degrees.csv). It is often formalized as a function $P(d)$ that, for any degree value d, gives the number of nodes of degree d divided by the number of nodes in the graph.
A graph is called a *scale-free graph* if the degree distribution obeys a power law: $P(d) \sim d^{-\gamma}$, where γ is a constant. Airport graphs are supposedly scale-free. Determine whether this airport graph is scale-free by plotting the histogram for $P(d)$ and visually determining if $\log\, P(d)$ appears to be a linear function of d.

(e) A *hub node* is a node with high node degree, sometimes defined as being among the top 10% of nodes when they are ordered by their degree values. Identify the hub nodes of the graph, using this definition.

(f) Algorithm design (and interview questions) often requires order-of-magnitude estimates. Estimate how many flights there are every day between the 321 major airports. Using the 5th column of the airport_connections.csv table, altogether the 4271 links between them are covered by airlines a total of 18334 times (i.e., each link is covered by approximately 4 airlines). However, some links are covered multiple times each day, some covered once every few days. Give an order-of-magnitude estimate for the number of flights each day.

(g) Assuming there are about 100 airplane disasters every year (for flights between major airports), based on the previous question, roughly what is the probability of a flight being a disaster? (Interview question)

4. **Articulation Points and Bridges**
An important property of an airport network is that it be failure-tolerant. If one airport shuts down, ideally the network should remain connected.

An *articulation point* is a vertex of a connected, undirected graph $G = (V, E)$ whose removal results in G becoming disconnected. A *bridge* of G is an edge whose removal disconnects G. A *simple cycle* of G is a sequence of distinct vertices $(v_1, \ldots, v_k)$ $(k > 1)$ for which each edge (v_i, v_{i+1}) is in E, and (v_k, v_1) is in E. (Assume there are no *self-loops* (edges from a node v to v) in this problem.) A *biconnected component* of G is a maximal set of edges, in which every pair of edges lies on a common simple cycle.

A *depth-first search tree* of G with starting vertex s is a directed graph $T = (V, E_T)$ recording DFS visitation of vertices. (DFS trees are defined in [KT] p.84.) For any edge $u \to v$ in a directed graph, u is the *parent*, and v is the *child*. Also if there is a directed path from u to v, u is an *ancestor* of v and v is an *descendant* of w. *Every vertex is both an ancestor and descendant of itself.* A *proper ancestor* of v is an ancestor that is not v.

(a) Prove that s is an articulation point of G if and only if it has two or more children in T.

(b) Prove that $v \neq s$ is an articulation point of G if and only if it has a child w such that there is no back edge from w or any descendant of w to a proper ancestor u of v.
(If such a node w exists, and u is the proper ancestor of v (so $u \neq v$), there is a path $u \to \cdots v \to \cdots w$ in the DFS tree T, as well as a path from w ending in a back edge to u, giving a simple cycle $u \ldots v \ldots w \ldots u$ in G. This cycle prevents articulation, since w still has a path back to u if v is removed.)

(c) Show how to compute all articulation points in $O(E)$ time.

(d) Prove that an edge $e \in E$ is a bridge if and only if it does not lie on any simple cycle of G.

(e) Find all articulation points and *bridges* in the airport graph. (See articulation_points.csv.) Which articulation points are hub nodes?