The Celebrity Problem

or, Is the correctness proof convincing?

March 19, 2006

Among n people $\{1, 2, \ldots, n\}$, a *celebrity* is defined as someone who is known to everyone, but who knows no one. The CELEBRITY PROBLEM is to identify the celebrity, if one exists, by asking only questions of the following form: "Excuse me, do you know person x?"

Mathematical formulation. Let $G = (V, E)$ be a directed graph (represented by, say, its incident matrix). There is a vertex for each of the n people, and an edge from u to v if person u knows person v. We define a *sink* of a directed graph to be a vertex with indegree $n - 1$ and outdegree 0. A celebrity corresponds to a sink of the graph. We note that a graph can have at most one sink.

Brute-force solution. The graph has at most $n(n-1)$ edges, and we can compute it by asking a question for each potential edge. At this point, we can check whether a vertex is a sink by computing its indegree and its outdegree. This brute-force solution asks $n(n-1)$ questions. Next we show how to to do this with at most $3(n-1)$ questions and linear place.

An elegant solution. Our algorithm consists of two phases: in the *elimination phase*, we eliminate all but one person from being the celebrity; in the *verification phase* we check whether this one remaining person is indeed a celebrity.

The elimination phase maintains a list of possible celebrities. Initially it contains all n people. In each iteration, we delete one person from the list. We exploit the following key observation: *if person 1 knows person 2, then person 1 is not a celebrity; if person 1 does not know person 2, then person 2 is not a celebrity.* Thus, by asking person 1 if he knows person 2, we can eliminate either person 1 or person 2 from the list of possible celebrities. We can use this idea repeatedly to eliminate all people but one, say person ℓ.

We now verify by brute force whether ℓ is a celebrity: for every other person i, we ask person ℓ whether he knows person i, and we ask persons i whether they know person ℓ. If person ℓ always answers no, and the other people always answer yes, the we declare person ℓ as the celebrity. Otherwise, we conclude there is no celebrity in this group.

Correctness. During the elimination phase, we maintain the invariant that is there exists a celebrity, then the celebrity is on the list. We can prove this by induction on the number of iterations. Thus, when elimination phase ends, either person ℓ is a celebrity or there is no celebrity.

Analysis. The elimination phase requires exactly $n - 1$ questions, since each question reduces the size on the list by 1. In the verification phase, we ask ℓ $n - 1$ questions, and we we ask the other $n - 1$ people one question. This phase requires at most $2(n - 1)$ questions, possibly fewer is ℓ is not a celebrity. So the total number of questions is $3(n - 1)$.

To efficiently implement the elimination phase, we maintain a queue that contains the remaining celebrities. Initially, we insert all n people to the queue. At each iteration we remove the top two elements off the queue, say v and w, and ask v whether he (or she) knows w. Depending on the outcome, we either insert v or w at the end of the queue. Each queue operation rakes $\Theta(1)$ time, so the whole process takes $\Theta(n)$ time.

An even better solution. We note that it is possible to save an additional $\lfloor \log_2 n \rfloor$ questions in the verification phase by not repeating any questions we already asked during the elimination phase. By maintaining the elements in a queue, the celebrity is involved (i.e., either asked or asked about) at least $\lfloor \log_2 n \rfloor$ questions during the elimination phase. This explains why we chose a queue instead of a stack.

Also, it is not hard to see that any algorithm must ask at least $2(n-1)$ questions if there exists a celebrity, since we must verify that the celebrity does not know anyone, and that everyone knows the celebrity.