Homework 2

Simon Zheng
260744353

October 13th, 2017

1

There must be at least two new edges. If we have a strongly connected directed graph G, and we add a vertex u such that we get $G \cup \{u\}$, then there must be a new edge going from u to any vertex in G, and an edge going from any vertex in G to u. If we only have the former, then we cannot reach u, thus not being a strongly connected graph anymore. If we only have the latter, then we cannot reach any vertex in G from u, and it is not a connected graph anymore either. If we have both, then we can reach u from any vertex in G, guaranteed by the incoming edge, and we can reach any vertex in G from u, guaranteed by the outgoing edge. It is guaranteed, as since we know G is strongly connected, then whichever vertex u goes to, we can go to anywhere else, and whichever vertex goes to u, we can get to from anywhere else.

2

We can represent this situation as a directed graph:

Each person is a vertex;

If person u knows person v, then there exists an edge going from u to v.

As such, a *celebrity* is a vertex c with an outdegree of 0 and an indegree of $|V| - 1$, where $|V|$ is the total number of vertices (people).

Now, the question forces us to ask every single person at least once to know who they know. This means visiting every vertex at least once (no matter if they are connected or not as we have to ask to know it), so the algorithm will be at least $O(n)$. We know the celebrity must know no one. With this, we can immediately exclude a vertex (person) as a celebrity the moment they know someone. But we must still know who else they know, as to identify the celebrity who is known by everyone. Fortunately, every time we visit a vertex (ask a person who they know), we can exclude anyone who wasn't previously mentioned, as the celebrity must be known by all.

A good way would be to maintain a single collection of people who are known by every single person we have visited. It must *only* include people who do not know anyone and is known by every previous person asked. As such, a good data structure would be a hash table, as we will be doing a lot of searches and deletions.

1. Starting from any person, for every other person, ask if they know them. If they do not know anyone, add to set of potential celebrities. Otherwise, add every person they know.

2. Now, for every following person, if they do not know anyone and are not included in the list of potential celebrities, then we are done, as we know there are no celebrities since we have a person who is not a celebrity and cannot know them.

3. If they do not know anyone but are a potential celebrity, then simply go to the next person.

4. If they know anyone and are a potential celebrity, remove them from the set of potential celebrities.

5. Now remove any person in the set that is not mentioned by the current person being asked. If the set becomes empty then we are done, and there are no celebrities.

6. Otherwise, continue until no one is left to ask. Either there will be no celebrity or a single person who is the celebrity in the set, as it is impossible to have more than 1 by virtue of having to be know by *every other person*.

Algorithm: BDFS(Vertex u)

```
Input: A vertex $u$ in a graph $G$
Output:
```

3

4

5