Question 3: Computing the girth of a graph

December 31, 2003

We consider here the problem of computing the *girth*, $g(G)$, of connected undirected graph $G = (V, E)$, i.e., the minimum length of a cycle (contained) in a graph G, or infinity if G has no cycle.

Throughout this notes, the term *cycle* refers to a simple closed walk and the term *path* refers to a simple non-closed walk. The usual BFS algorithm ignores vertices that have already been explored. On the other hand, if we reach an already-explored vertex at depth r, then G must contain a cycle of size $\leq r$; that is, the girth of G is at most r. Conversely, if G contains an r-cycle C and we start the BFS algorithm at some vertex $v \in V(C)$, then we are guaranteed to reach an already-explored vertex by the rth stage. Therefore, we can compute the girth $g(G)$ of G as

$$\min_{v \in V(G)} \left(\begin{array}{l} \text{minimum depth at which a vertex appears for the} \\ \text{second time, when we run a BFS starting at } v \end{array} \right).$$

We now present an algorithm to compute the girth of a graph. Our algorithm will use a BFS approach from each vertex of the graph.

When searching from vertex v - that is, constructing a rooted tree with root v - we need to keep track of the parent of each other vertex; otherwise we might mistakenly identify a closed walk as a cycle.

Algorithm GIRTH(G)

$g(G) \leftarrow \infty$ ► the size of the smallest cycle already found.
for every $v \in V(G)$ **do**
    $S \leftarrow \emptyset$; $R \leftarrow \{v\}$; Parent(v) $\leftarrow$ NULL; $D(v) \leftarrow 0$ ► $D(w) = d(v, w)$.
    while $R \neq \emptyset$ **do**
        choose $x \in R$
        $S \leftarrow S \cup \{x\}$; $R \leftarrow R \setminus \{x\}$
        for every $y \in N(x) \setminus \{\text{Parent }(x)\}$ **do** ► $N(u) = \big\{w \in V \mid (u, w) \in E\big\}$.
            if $y \notin S$ **then**
                Parent $(y) \leftarrow x$
                $D(y) \leftarrow D(x) + 1$
                $R \leftarrow R \cup \{y\}$
            else
                $g(G) \leftarrow \min\big\{g(G),\ D(x) + D(y) + 1\big\}$
return $g(G)$

Clearly, the running time of this algorithm is $O\big(V(V + E)\big) = O(VE)$.