— MPRI —
Examen "Graphs and Algorithms"
February 13, 2007 — 8h45-11h45

This examen has two parts, with weights (time and grade) 2/3 for Part I and 1/3 for Part II. No electronic communication devices (e.g., portable computer, mobile phone, etc.) are allowed. Lecture notes (paper) are however authorized.

Part I. Labeling, routing, and augmentation in trees

Let $T = (V, E)$ be a tree of $n \geq 1$ nodes. Nodes of T have distinct IDs that are integers in $\{1, \ldots, n\}$. The ID of node $u \in V$ is denoted by $\mathrm{ID}(u)$. The $k = \deg(u)$ edges incident to node $u \in V$ are labeled from 1 to k. These labels are called port numbers. The label of edge $e = \{u, v\} \in E$ is $\mathrm{port}_u(e)$ at u, and $\mathrm{port}_v(e)$ at v. We may have $\mathrm{port}_u(e) \neq \mathrm{port}_v(e)$.

The exercise considers different labeling schemes for different objectives : adjacency, routing, augmentation, etc.

Two nodes u and v are said adjacent if and only if $\{u, v\} \in E$. We consider the following labeling scheme. T is rooted at an arbitrary node $r \in V$. The label $\mathcal{L}_{adj}(r)$ of r is the pair $(\mathrm{ID}(r), 0)$. The label of $u \neq r$ is $\mathcal{L}_{adj}(u) = (\mathrm{ID}(u), \mathrm{ID}(v))$ where v is the parent of u in T.

Question 1. What is the maximum size (in bits) of a label $\mathcal{L}_{adj}(u)$? Show how, given the labels $\mathcal{L}_{adj}(u)$ and $\mathcal{L}_{adj}(u')$ of two nodes u and u', one can determine whether u and u' are adjacent or not.

Recall that a *center* of an n-node tree T is a node u such that all trees resulting from removing u from T have size at most $n/2$.

Question 2. Prove that a tree has at least one center. (Hint : consider the procedure that, starting at a leaf, travels into the tree looking for a center, and stops when it finds one).

We define the labeling $\mathcal{L}_{anc}$ recursively as follows. Let c be a center of T. For any $u \in V$, we set $c_u^{(0)} = c$, $T_u^{(0)} = T$, and define $T_u^{(1)}$ as the subtree of $T \setminus \{c_u^{(0)}\}$ containing u. Then, let $c_u^{(1)}$ be the center of $T_u^{(1)}$ closest to c, and let $T_u^{(2)}$ be the subtree of $T_u^{(1)} \setminus \{c_u^{(1)}\}$ containing u. And so on. One constructs in this way two sequences $(T_u^{(0)}, T_u^{(1)}, \ldots, T_u^{(q_u)})$ and $(c_u^{(0)}, c_u^{(1)}, \ldots, c_u^{(q_u)})$ where :

1. $T_u^{(0)} = T$;
2. $c_u^{(i)}$ is the center of $T_u^{(i)}$ closest to c in T ;
3. $T_u^{(i+1)}$ is the subtree of $T_u^{(i)} \setminus \{c_u^{(i)}\}$ containing u ;
4. $c_u^{(q_u)} = u$.

A node $c_u^{(i)}$ for some index $i \geq 0$ and some node $u \in V$ is called a center of level i.

Question 3. Draw the complete binary tree of depth 3 (and hence of 15 nodes), and the path of 16 nodes, in which the centers of level 0, 1, 2, and 3 are indicated. Prove that for any $u \in V$, we have $q_u \leq \lceil \log_2 n \rceil$. Give a tree for which this bound is reached for at least one node u.

For $u \in V$ we set $\mathcal{L}_{route}(u) = (\mathrm{ID}(c_u^{(0)}), \mathrm{ID}(c_u^{(1)}), \dots, \mathrm{ID}(c_u^{(q_u)}))$. For two nodes u and v, let $j \geq 0$ be the largest index such that $\mathcal{L}_{route}(u)$ and $\mathcal{L}_{route}(v)$ share a same prefix of length $j+1$, i.e., there exists $(x_0, \dots, x_j)$ that is prefix of both $\mathcal{L}_{route}(u)$ and $\mathcal{L}_{route}(v)$.

Question 4. Prove that the shortest path between u and v goes through the node w such that $\mathrm{ID}(w) = x_j$.

We now consider augmented trees for analyzing greedy routing using Kleinberg framework. The model that we consider is the following. Each node $u \in V$ is given a long range contact $\mathrm{lrc}(u) \in V$, and the arc $(u, \mathrm{lrc}(u))$ is added to the tree. Greedy routing performs as follows. Let $\deg(u)$ be the number of neighbors of node u in T, and let $\mathrm{dist}(u, v)$ be the distance between u and v in T, i.e., the minimum number of edge traversals for traveling between u and v in T. To perform greedy routing, a node u holding a message for target t considers its neighbors $v_1, \dots, v_{\deg(u)}$ in T and its long range contact $\mathrm{lrc}(u)$, and forwards the message to the node $w \in \{v_1, \dots, v_{\deg(u)}\} \cup \{\mathrm{lrc}(u)\}$ that is closest to t in T, i.e., $\mathrm{dist}(w, t) = \min\{\mathrm{dist}(v_1, t), \dots, \mathrm{dist}(v_{\deg(u)}, t), \mathrm{dist}(\mathrm{lrc}(u), t)\}$.

In this exercise, we pick, for every node u, the contact $\mathrm{lrc}(u)$ uniformly at random in $\{c_u^{(0)}, c_u^{(1)}, \dots, c_u^{(q_u)}\}$ where this sequence is defined in Question 3.

Question 5. Let t be the target node in T, and let u be the current node. Let $j \geq 0$ be the largest index such that $\mathcal{L}_{route}(u)$ and $\mathcal{L}_{route}(t)$ share a same prefix of length $j+1$. What is the probability that the long range contact of u is $c_t^{(j)}$? What does greedy routing perform at u if $\mathrm{lrc}(u) = c_t^{(j)}$?

Question 6. Let s and t be two nodes in T. What is the expected number of steps of greedy routing from s to t in T augmented as above ?

Finaly, we focus on the design of shortest path routing schemes in (non augmented) trees. Precisely, let $\textsc{name}(u)$ and $\textsc{table}(u)$ be respectively the name of node u, and the routing table stored at node u. We have to define the names and tables of the nodes, and a routing function $\texttt{routing}(\textsc{name}(v), \textsc{table}(u))$ that takes as input the name of the destination v and the routing table of the current node u, and returns the port number $p \in \{1, \dots, \deg(u)\}$ of the incident edge of u through which messages to v must be forwarded. (Note that port numbers are fixed in this exercise, and the routing scheme cannot modify these port numbers, in contrast with the routing scheme described in the course which used specific port numbers set by the designer).

Question 7. Let S_n be the star of n nodes. Assume the center c is given name $\textsc{name}(c) = (\mathrm{ID}(c), 0)$, and for every leaf v, $\textsc{name}(v) = (\mathrm{ID}(v), p)$ where $p = \mathrm{port}_c(\{c, v\})$. Describe a shortest path routing scheme in S_n using tables of $\lceil \log_2 n \rceil$ bits.

Question 8. Adapt the labeling $\mathcal{L}_{route}$ to design a shortest path routing scheme in any n-node tree using names on $O(\log^2 n)$ bits, and local routing tables of size $O(\log^2 n)$ bits.

Part II.