Instructor: Bill Lenhart 1 Due: Noon, 22 February

Traversals + Counting = Solution

**Question 1** (KT 3.9). There’s a natural intuition that two nodes that are far apart in a communication network—separated by many hops—have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here’s one that involves the susceptibility of paths to the deletion of nodes.

Suppose that an \(n\)-node undirected graph \(G = (V, E)\) contains two nodes \(s\) and \(t\) such that the distance between \(s\) and \(t\) is strictly greater than \(n/2\). Show that there must exist some node \(v\), not equal to either \(s\) or \(t\), such that deleting \(v\) from \(G\) destroys all \(s - t\) paths. In other words, the graph obtained from \(G\) by deleting \(v\) contains no path from \(s\) to \(t\). Give an algorithm with running time \(O(m + n)\) to find such a node \(v\). Describe your algorithm in prose and prove that it works correctly. Make sure your algorithmic description is clear and concise. As a hint, imagine performing a breadth-first search from \(s\). How large is each layer \(L_i\) along the way to \(t\)？

Tree Traversal & Graph Metrics

It might seem odd to apply BFS or DFS to a tree—you just get the tree you started with! However, the traversal of the tree in a particular order can allow for efficient computation of useful quantities. To see this, let’s define some quantities.

**Definition 1.** The distance between vertices \(u\) and \(v\) in a graph \(G = (V, E)\), denoted \(\text{dist}(u, v)\), is given by \(\text{dist}(u, v) = \min\{k : \text{there is a } u - v \text{ path of length } k\}\).

**Definition 2.** The diameter of a graph \(G\), denoted \(\text{diam}(G)\), is given by \(\text{diam}(G) = \max\{\text{dist}(u, v) : u, v \in V\}\).

That is, the diameter of \(G\) is the distance between two “farthest-apart” vertices in \(G\).

**Question 2.** Let \(T = (V, E)\) be a tree having \(n\) vertices. Give a linear time, that is \(O(n)\)-time, algorithm to find the diameter of \(T\). Prove that your algorithm is correct. Suggestion. Consider modifying recursive DFS so that it also computes, for each vertex \(v\)

(a) The diameter of the subtree of \(T\) rooted at \(v\)

(b) The longest path from \(v\) to a leaf in the subtree of \(T\) rooted at \(v\)

Then show how, knowing this information for all children of some vertex \(u\), we can determine this information for \(u\) itself. (Why do you think we need to know both (a) and (b)?)

**Question 3.** Until recently, there was no known method for computing the diameter of a graph that didn’t first compute the shortest path between all pairs of nodes. When graphs are dense, all-pairs shortest paths is fairly expensive, so some people have explored algorithms that can more quickly estimate the diameter of the graph.

Develop a linear-time algorithm\(^1\) that, given a graph \(G\), returns a diameter estimate that is always within a factor of 1/2 of the true diameter. That is, if the true diameter is \(d = \text{diam}(G)\) then you should return a value \(k\) where \(d/2 \leq k \leq d\).

Applications of Traversals

**Question 4.** Answer Question 11 from Chapter Three of your text Algorithm Design. In addition to designing the algorithm, justify its correctness and time and space requirements. Suggestion. Consider first instances of the problem such that no two communication events happen at the same time, then think about how you might modify your solution to deal with multiple events happening at the same time.

**Question 5.** Answer Question 12 from Chapter Three of your text Algorithm Design. In addition to designing the algorithm, justify its correctness and time and space requirements. Hint: Create a graph with 2 vertices for each person \(P_i\); call the vertices \(b_i\) and \(d_i\). The vertices represent the (unknown) dates that person \(P_i\) was born and died. Now add edges as follows

---

\(^1\)See Problem 2 for a precise definition of the distance between \(u\) and \(v\).

\(^2\)Recall that the length of a path \(P\) between two vertices \(u\) and \(v\) is the number of edges on the path.

\(^3\)Remember: Linear time for a graph \(G = (V, E)\) means time \(O(n + m)\), where \(n = |V|\) and \(m = |E|\).
• For each person $P_i$, add edge $(b_i, d_i)$ (this says $P_i$ was born before $P_i$ died)
• For each rule “$P_i$ died before $P_j$ was born” add edge $(d_i, b_j)$
• For each rule “the lifespans of $P_i$ and $P_j$ overlapped at least partially” add edges $(b_i, d_j)$ and $(b_j, d_i)$.

Now prove that the graph constructed above represents a consistent set of rules if and only if the graph does not contain any cycles.

Optional Challenge Problem!

**Question 6.** There’s another well-known algorithm to compute the diameter of a tree that may surprise you. Here it is

• Pick any vertex $v$ in $T$
• Find a vertex $u$ that maximizes $\text{dist}(v, u)$, by using a traversal
• Find a vertex $w$ that maximizes $\text{dist}(u, w)$, again, using a traversal
• Return $\text{dist}(u, w)$ as the diameter of $T$.

Prove that this algorithm is correct. Suggestion: Draw some pictures. Consider the sub-tree $T'$ that consists of the (unique!) paths from $v$ to $u$ and $u$ to $w$. Imagine some longer path exists—how does it connect to $T'$?