Problem set 7 - Exercise 2

Fabio Pernisi

January 2024

Problem Description

Design a data structure that, given a rooted tree T on n nodes, preprocesses it in $O(n \log n)$ time and then can answer in $O(\log n)$ time queries of the form: "Given $a \in T$ and $k \in \mathbb{N}$, what is the ancestor of a located k levels above it?" That is, when $k = 1$ we ask for the parent of a, when $k = 2$ we ask for the parent of the parent of a, and so on.

Hint: Consider intervals [dfs-in-time, dfs-out-time] for each level of the tree.

Solution

We propose a data structure that preprocesses a rooted tree T using Depth-First Search (DFS) to enable efficient ancestor queries.

Preprocessing

- Perform a DFS traversal of T, recording the dfs-in-time and dfs-out-time for each node.
- Construct a dictionary mapping each depth to the list of nodes at that depth, including their dfs time intervals.

Querying To find the ancestor of a node a at depth D located k levels above it, perform the following steps:

1. Calculate the target depth $D' = D - k$.
2. Retrieve the list of nodes at depth D' from the dictionary.
3. For each node b at depth D', check if the dfs interval of a is contained within the dfs interval of b. If so, b is the ancestor we are looking for.
4. Return the ancestor node b.

Pseudocode

```
Input: Tree $T$ rooted at $r$, node $a$, integer $k$
Output: Ancestor of $a$ located $k$ levels above
procedure PreprocessTree($T, r$)
    $depthDict \leftarrow$ empty dictionary
    Perform DFS on $T$, filling $depthDict$ and recording dfs times
end procedure
function FindAncestor($a, k$)
    $D \leftarrow$ depth of $a$
    $D' \leftarrow D - k$
    $ancestors \leftarrow depthDict[D']$
    for $node$ in $ancestors$ do
```

```
        if dfs interval of $a$ is within dfs interval of $node$ then
            return $node$
        end if
    end for
end function
```