Exercise Session 24/05/2023

Algorithms and Data Structures

Exercise 179 (f16)

Write an algorithm BTreePrintRange(T, a, b) that, given a B-tree T and two values a < b, prints all the keys k in T that are between a and b, that is, a < k < b. Recall that a node x in a B-tree has the following properties:

- x.n is the number of keys
- x.key[1] ≤ x.key[2] ≤ ... ≤ x.key[x.n] are the keys
- x.leaf tells whether x is a leaf
- x.c[1], x.c[2], ... , x.c[x.n + 1] are the pointers to x's children

Exercise 161 (f15)

A breadth-first search over a graph G returns a vector π that represents the resulting breadth-first tree, where the parent π[v] of a vertex v is the next-hop from v on the tree towards the source of the breadth-first search.

Question 1: Write an algorithm BFS-First-Common-Ancestor(π, u, v) that finds the first common ancestor of two given nodes in the breadth-first tree, or null if u and v are not connected in G. The complexity of BFS-First-Common-Ancestor must be O(n). Briefly analyze the space complexity of your solution

Question 2: Write an algorithm BFS-First-Common-Ancestor-2(π, D, u, v) that is also given the distance vector D resulting from the same breadth first search. BFS-First-Common-Ancestor-2 must be functionally equivalent to BFS-First-Common-Ancestor (as defined in Exercise 1) but with space complexity O(1).