NTIN066 - practicals 2

David Mareček

February 23, 2022

1. What is the worst-case complexity of the SUCC function from your first assignment?

2. What is the average complexity of SUCC across all the keys in a tree?
Consider enumeration of all keys in a binary search tree using MIN and n times SUCC. Prove that although a SUCC requires $\Theta(\log(n))$ time in the worst case, the whole enumeration takes only $\Theta(n)$ time. Can we formulate any amortization argument?

3. Show how to make a BST perfectly balanced in $\Omega(n)$ time.
How to manage it with as little additional memory as possible? It is easy to prove it for $O(n)$, harder for $O(\log(n))$, but it is also possible to use only $O(1)$ additional memory.

4. Show that either Insert or Delete in a perfectly balanced BST tree must have worst-case time complexity $\Omega(n)$.
This is easy for perfectly balanced trees on $n = 2^k - 1$ vertices, whose shape is uniquely determined.

5. Add a Delete operation into lazily balanced trees
Use the same Delete as iplemented in ordinary BSTs. Use the same potential to analyze it.

6. What would go wrong if we forgot to add the exception for difference 1 in the definition of the potential $\Phi(v)$ in lazily balanced trees?

7. Decrement in Binary counter
What goes wrong with the amortized complexity if we also want to decrement? Could we improve the counter so that the amortized complexity is $O(1)$ for each possible sequence of increments and decrements?