Due: Tuesday, March 7, 2023, 11:59pm

Submit online at (clickable link): `https://forms.gle/n1R6yn5gKrzDW1sx9`

1 Chang's Folly

In a previous semester, Prof. Chang suggested in lecture that as operations are applied to a splay tree, it becomes more and more balanced. Show that this is not necessarily true by providing a sequence of `search()` operations that will turn any n node splay tree into a right-going chain. That is, in the resulting tree, all of the nodes are along the path starting at the root that keeps visiting the right child.

Briefly justify your answer.

Why does your example not contradict the amortized analysis?

2 Alternate Splay Tree Insertion

Here is an alternate implementation of insert for Splay trees. First, insert x as you would normally do in a binary search tree. After this insertion x becomes a leaf. Then, rotate x as you would in the splay operation until x is the root of the tree.

This implementation is actually different from the one discussed in lecture. In this implementation, x becomes the root of the tree, which is not the case in the original implementation.

Fortunately, we do not have to redo the entire amortized analysis of Splay trees for this change, because the amortized analysis is valid as long as each node in the tree has enough credits (i.e., as long as the invariant holds). It does not matter that the tree was not constructed using splay tree operations. We just have to make sure that every node y still has $\text{rank}(y)$ credits after we insert x at the leaf (and before we run the splay operation).

1. After inserting x as a leaf in the alternate implementation of insert, which nodes might see an increase in their size?

2. How many nodes are there that see an increase in their size? Briefly justify your answer. *Note: recall that the height of a splay tree could be linear.*

3. How many nodes are there that see an increase in their rank? Briefly justify your answer. *Note: it is possible that a node's size increases, but its rank does not.*

4. If a node's rank does increase, how much can the rank change? Briefly justify your answer.

5. How many credits would you need to add to make sure that every node y still has $\text{rank}(y)$ credits? Briefly justify your answer.

6. What is the amortized running time of the entire alternate insert operation (including inserting x at the leaf and splaying at x)? Briefly justify your answer.

3 Tall Skinny Fibonacci Heap

Specify a sequence of Insert, Decrease-Key and Extract-Min operations that would create a Fibonacci Heap with 4 nodes in a single chain:

Show what your Fibonacci Heap looks like at important steps (i.e., you can bunch together the boring parts). Use actual numbers for the items.