Homework 2

ECE345 - Group 16

October 13th, 2023

Total pages: TBD

Team Member	Student Number
Ardavan Alaei Fard	1007934620
Rowan Honeywell	1007972945
Isaac Muscat	1007897135

Contents

Question 1

(a)

(b)

(c)

Question 2

(a)

(b)

Question 3

(a)

(b)

(c)

Question 4

Note: Leftist heap will be abbreviated as LH.

(a)

Let s represent the smallest complete sub-tree of an LH L starting from the root. Since the rank of L will be the length of the shortest path from the root to the leaf, the height of s will have a height of the rank of the root of L (otherwise the s would not be complete). If m is the number of nodes in s, the height of s will be $\mathcal{O}(\lg m)$ which will be the same as the rank of the root. If n is the number of nodes in L, then since s is a sub-tree of L, $n \geq m \implies \lg n \geq \lg m \implies$ the rank of the root of an LH is $\mathcal{O}(\log n)$. QED

(b)

From (a), we know that the rank of the root of an LH is $\mathcal{O}(\log n)$ which is the same as the length of the rightmost path. We also know that to merge two sorted sequences using **MERGE** (CLRS, 4th, page 38), it takes $\Theta(n)$. If the size of two leftist heaps l_1 and l_2 have sizes n_1 and n_2, then to merge the rightmost paths of l_1 and l_2, the **MERGE** procedure will have to iterate over $\mathcal{O}(\log n_1) + \mathcal{O}(\log n_2) = \mathcal{O}(\log n)$ elements. Therefore, to merge l_1 and l_2, it takes $\mathcal{O}(\log n)$ time. To show that the order invariant is maintained, suppose that an LH l_1 with $rank = 0$ is being added to the right child of the root of another LH l_2 with its right child removed in the LH merge procedure where merging two LHs splits both of them into sub-trees with their root's right child removed. Since the key of the root of l_1 is larger and all other nodes of l_1 are larger than its root by the definition of an LH, all other nodes in l_1 will be larger than the root of l_2. QED

(c)

Test

(d)

(e)

To implement **DeleteMin** and Insert, we can utilize the **Merge** procedure that runs in $\mathcal{O}(\log n)$ time.

$\textsc{DeleteMin}(H)$

$l = \textsc{Leftist-Heap}(H.\mathit{root}.\mathit{left})$

$r = \textsc{Leftist-Heap}(H.\mathit{root}.\mathit{right})$

$H = \textsc{Merge}(l, r)$

$\textsc{Insert}(H, i)$

$H = \textsc{Merge}(H, \textsc{Leftist-Heap}(H.i))$

References

[1] CLRS, a, b, c, d