An Introduction to and Analysis of the Hollow Heap Data Structure with a Comparison to the Fibonacci Heap Data Structure

Alisha Sprinkle and Courtney Dixon

December 8, 2019

1 Abstract

"For the paper, I would like you to explain the data structure clearly and the amortized analysis. Try to use the accounting method as sometimes that is easier to visualize what is happening in the operations. For the extension, you may implement this data structure, then implement Fibonacci heaps and compare the two in terms of practical efficiency."

2 Introduction

Little bit of this and that talk

3 Hollow Heap: the data structure

"Clearly explain the data structure"

4 Analysis

"Amortized Analysis via Accounting Method"

5 Hollow Heap Implementation

5.1 Methods

□ makeHeap()

□ insert(HollowNode e, int key, HollowHeap h)

□ meld(HollowHeap g, HollowHeap h)

□ findMin(HollowHeap h)

□ decreaseKey(HollowNode e, int key, HollowHeap h)

□ deleteMin(HollowHeap h)

□ delete(HollowNode e, HollowHeap h)

□ doRankedLinks(HollowNode u)

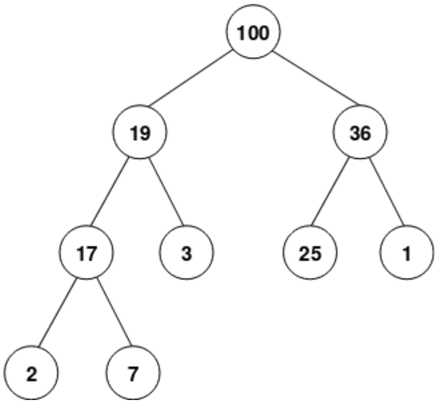
□ doUnrankedLinks()

□ makeNode(HollowNode e, int key)

□ link(HollowNode v, HollowNode w)

□ addChild(HollowNode v, HollowNode w)

5.2 Pictures



6 Fibonacci Heap Implementation

6.1 Methods

- □ FibonacciHeap()
- □ insert(Node newNode)
- □ cut(Node cutNode, Node fromNode)
- □ cascadeCut(Node cutNode)
- □ makeChild(Node child, Node parent)
- □ decreaseKey(Node decNode, int ket)
- □ findMin()
- □ delete(Node dNode)
- □ consolidate()
- □ displayHeap()

6.2 Pictures

7 Hollow versus Fibonacci: a comparison

“Compare the two data structures in terms of practical efficiency”

7.1 Similarities

1.
2.
3.

7.2 Dissimilarities

1.
2.
3.

References

[1] Thomas Dueholm Hansen et al. “Hollow Heaps”. In: *CoRR* abs/1510.06535 (2015). arXiv: `1510.06535`. URL: `http://arxiv.org/abs/1510.06535`.