Something about Computer Science

Karsken Bælg, 20051234
Karsken Bælg, 20051234

Master's Thesis, Computer Science
October 2013
Advisor: Anders Møller

AARHUS
UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Contents

Chapter 1

Introduction

needs content

Chapter 2

Binary heaps

needs content

2.1 Binary heap with array

needs content

2.2 Implementing decrease key

needs content

2.3 Time-complexity for binary heap with array

2.4 Binary heap with pointers

needs content

2.5 Implementing decrease key

needs content

2.6 Time-complexity for binary heap with pointers

2.7 Testing correctness of Binary Heaps

needs content

Chapter 3

Fibonacci heaps

In this chapter we focus on Fibonacci heaps, which is a data structure that has a forest of rooted trees as opposed to a binary heap that only has one tree [1]. The data structure was invented by Michael L. Fredman and Robert Endre Tarjan and was published in the Journal of ACM in 1987. It has it name because the size of any subtree in a Fibonacci heap will be lower bounded by F_{k+2} where k is the degree of the root and F is the Fibonacci function. Below is the time-complexities of each of the heap operations listed:

Operation	Binary heap	Fibonacci heap v1 (amortized)	Fibonacci heap v2 (amortized)
MakeHeap	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
FindMin	$\Theta(1)$	$\Theta(1)$	$O\left(l(\lg(\frac{n}{l})+1)\right)$
Insert	$\Theta(\lg n)$	$\Theta(1)$	$\Theta(1)$
DeleteMin	$\Theta(\lg n)$	$O(\lg n)$	$\Theta(1)$
DecreaseKey	$\Theta(\lg n)$	$O(1)$	$O(1)$
Delete	$\Theta(\lg n)$	$O(\lg n)$	$\Theta(1)$
Meld	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$

3.1 Fibonacci heap version 1

The first Fibonacci heap variant we present is the original version proposed in FT87. A potential function is used to analyze the perfomance, thus the above stated time-complexities are amortized.

3.2 Worst case time-complexity for fib-v1

There are three operations where changes to the potential occurs, and thus, the stated times are amortized for DeleteMin, DecreaseKey and Delete. Below we show the worst-case for each of these operations:

3.2.1 Worst case time-complexity for DeleteMin

3.3 Fibonacci heap version 2

needs content

3.4 Worst case time-complexity for fib-v2

needs content

3.5 Testing correctness of Fibonacci Heaps

needs content

Chapter 4

Test-results

needs content

Chapter 5

Dijkstra

needs content

Chapter 6

Binary heap vs Fibonacci heap

needs content

Chapter 7

Test-results

needs content

Chapter 8

Conlusion

needs contents

Bibliography

[1] Robert Endre Tarjan Michael L. Fredman and. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.