	Linked list	Sorted array	Heap	Optimized heap
`insert()`	$\Theta(1)$	$\Theta(n)$	$\Theta(\lg n)$	$\Theta(\lg n)$
`min()`	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
`extractMin()`	$\Theta(n)$	$\Theta(1)$	$\Theta(\lg n)$	$\Theta(\lg n)$
`decreaseKey()`	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\lg n)$
`delete()`	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\lg n)$

	Linked list	Sorted array	Heap	Optimized heap	Fibonacci heap (amortized)
`insert()`	$\Theta(1)$	$\Theta(n)$	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(1)$
`min()`	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
`extractMin()`	$\Theta(n)$	$\Theta(1)$	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(\lg n)$
`decreaseKey()`	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\lg n)$	$\Theta(1)$
`delete()`	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\lg n)$	$\Theta(\lg n)$
`union()`	$\Theta(1)$	$\Theta(n)$	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(1)$

Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms

MICHAEL L. FREDMAN

University of California, San Diego, La Jolla, California

AND

ROBERT ENDRE TARJAN

AT&T Bell Laboratories, Murray Hill, New Jersey

Abstract. In this paper we develop a new data structure for implementing heaps (priority queues). Our structure, *Fibonacci heaps* (abbreviated *F-heaps*), extends the binomial queues proposed by Vuillemin and studied further by Brown. F-heaps support arbitrary deletion from an n-item heap in $O(\log n)$ amortized time and all other standard heap operations in $O(1)$ amortized time. Using F-heaps we are able to obtain improved running times for several network optimization algorithms. In particular, we obtain the following worst-case bounds, where n is the number of vertices and m the number of edges in the problem graph:

(1) $O(n \log n + m)$ for the single-source shortest path problem with nonnegative edge lengths, improved from $O(m \log_{(m/n+2)} n)$;
(2) $O(n^2 \log n + nm)$ for the all-pairs shortest path problem, improved from $O(nm \log_{(m/n+2)} n)$;
(3) $O(n^2 \log n + nm)$ for the assignment problem (weighted bipartite matching), improved from $O(nm \log_{(m/n+2)} n)$;
(4) $O(m\beta(m, n))$ for the minimum spanning tree problem, improved from $O(m \log \log_{(m/n+2)} n)$, where $\beta(m, n) = \min \{i \mid \log^{(i)} n \le m/n\}$. Note that $\beta(m, n) \le \log^* n$ if $m \ge n$.

Acta Informatica 13, 53 – 58 (1980)

Acta Informatica

Applications of Efficient Mergeable Heaps for Optimization Problems on Trees

Zvi Galil*

Department of Mathematical Sciences, Computer Science Division, Tel-Aviv University, Ramat-Aviv, Tel-Aviv, Israel

Summary. It is shown how to use efficient mergeable heaps to improve the running time of two algorithms that solve optimization problems on trees.

0. Introduction

A *tree* is a directed graph with a distinguished vertex – the *root* such that there is exactly one directed path from the root to any other vertex. If there is an edge