

LAB 4

吴佳遥 517370910257

2

Procedure	Time complexity
MakeHeap	$\Theta(1)$
Insert	$\Theta(1)$
Minium	$\Theta(1)$
ExtractMin	$\mathcal{O}(\log n)$
Union	$\Theta(1)$
DecreaseKey	$\Theta(1)$
Delete	$\mathcal{O}(\log n)$

3

FibHeap has a better amortized running time than the simple min-heap. For Insert, Union, DecreaseKey, it performs much better than min-heap. But for ExtractMin and Delete operation, since the worst case for min-heap is $\mathcal{O}(\log n)$, FibHeap may do worse than min-heap.

The advantage of FibHeap is that it performs well in UNION, merging two heaps together, and it has a better amortized running time in all operations besides Delete and ExtractHeap.

The disadvantage of FibHeap is that it works poorly in ExtractMin and Delete when the data becomes large. Its constant factors and programming complexity make it less practical in real-time applications.

4.

FibHeap is designed so that the maintenance only happens when deleting elements. Thus, operations without deleting elements will greatly be improved.

It may be preferred in circumstances that UNION operations are needed, in circumstances that the number of ExtractMin and Delete operations is small relative to the number of other operations performed.

And due to its constant factors and programming complexity, FibHeap will be more desirable in applications that manage large amounts of data.

Reference

Leiserson Charles E. Cormen Thomas H. and etc. *Introduction to Algorithms - Third Edition*. The MIT Press, 2009

