Binomial Queues

CSE 373

Data Structures

Lecture 12

Binomial Queues - Lecture 12 Reading

• Reading

Section 6.8,›

10/25/02

Merging heaps

• Binary Heap is a special purpose hot r› FindMin, DeleteMin and Insert only

› does not support fast merges of two heap

• For some applications, the items arrive prioritized clumps, rather than individu• Is there somewhere in the heap design we can give up a little performance so can gain faster merge capability?

Binomial Queues - Lecture 12 10/25/02

Binomial Queues

• Binomial Queues are designed to be quickly with one another

• Using pointer-based design we can mlarge numbers of nodes at once by simpruning and grafting tree structures

• More overhead than Binary Heap, but flexibility is needed for improved mergispeed

Binomial Queues - Lecture 12 10/25/02

Worst Case Run Time

Binomial QBinary Heap

(log N) (log N) Insert

O(log N)

(1) FindMin

(log N)

(log N) DeleteMin

O(log N)

(N) Merge

Binomial Queues - Lecture 12 10/25/02

Binomial Queues

(1) FindMin • Binomial queues give up

performance in order to provide O(log N) performance

• A **binomial queue** is a collection (or forheap-ordered trees

› Not just one tree, but a collection of tr› each tree has a defined structure and › each tree has the familiar heap-order

Binomial Queues - Lecture 12 10/25/02

Binomial Queue with 5 T*B4 B3 B2* depth

22 = 4

2

Binomial Queues - Lecture 12

23 = 8

3

24 = 16

4

number of elements

10/25/02

Structure Property

*B2*

2

22 = 4

• Each tree contains two

copies of the previous tree

› the second copy is attached at

the root of the first copy

• The number of nodes in a

tree of depth d is exactly 2d

depth

number of elements

Binomial Queues - Lecture 12 10/25/02

• Any number N can be representebase 2

› A base 2 value identifies the powerthat are to be included

Decimal10

| 3 | 4 | 5 |
| --- | --- | --- |
| 3 | 4 | 5 |
|  |  |  |
| 1 | 0 | 1 |
| 1 | 0 | 0 |
|  | 1 | 1 |
|  |  |  |

Binomial Queues - Lecture 12 Powers of 2 Hex16

**0**

**1**

**=**

**20**

**1**

**2**

**0**

**1**

**4**

**0**

**1**

**8**

**0**

**1**

**=**

**21**

**=**

**22**

**=**

**23**

10/25/02

Numbers of nodes

• Any number of entries in the binoqueue can be stored in a forest of

binomial trees

• Each tree holds the number of nodappropriate to its depth, ie 2d node

• So the structure of a forest of binotrees can be characterized with a binary number

1·22 + 0·21 + 0·20 = 4 nodes› 1002

Binomial Queues - Lecture 12 10/25/02

Structure Examples

22 = 4 4

4

8 5

8

7

22 = 4 **N=410=1002**

20 = 1 21 = 2 22 = 4 **N=210=102**

4

9 4

8 5

8

7

**N=510=1012**

20 = 1 21 = 2 22 = 4 **N=310=112**

What is a merge?

• There is a direct correlation between

› the number of nodes in the tree

› the representation of that number in base

› and the actual structure of the tree

• When we merge two queues, the numnodes in the new queue is the sum of • We can use that fact to help see how fmerges can be accomplished

Binomial Queues - Lecture 12 10/25/02

BQ.1

1 =  2

2 = 4 2

**N=110=12**

4

8

+ BQ.2

21 = 22 = 4 **N=210=102**

4 8 = BQ.3

21 =  22 = 4 **N=310=112**

Example 1.

Merge BQ.1 and

BQ.2

Easy Case.

There are no

comparisons and

there is no

restructuring.

1

3

4

6

1 =

21 =

21 =

2

1

3

2 = 4

22 = 4

22 = 4

BQ.1

Example 2.

2

**N=210=102**

Merge BQ.1 and BQ.2

This is an add with a

+ BQ.2 carry out.

It is accomplished with

**N=210=102**

one comparison and

one pointer change:

O(1)

4

= BQ.3

6

**N=410=1002**

1

3

BQ.1

1 =

2

2 = 4

2

**N=310=112**

Example 3.

4

6

+ BQ.2

Merge BQ.1 and BQ.2

the Part 1 - Form

21 =

22 = 4

**N=310=112**

carry.

7

= carry

8

21 =  22 = 4 **N=210=102**

1 7

3

+ BQ.1

8

carry

1 =  2

2 = 4 2

**N=310=112** 0 = 1 2

21 = 2

2 = 4 2

**N=210=102**

4

Example 3.

6

+ BQ.2

Part 2 - Add the existing

values and the carry.

21 = 22 = 4 **N=310=112**

7 1

8

3

4

= BQ.3

6

21 =  22 = 4 **N=610=1102**

Merge Algorithm

• Just like binary addition algorithm

• Assume trees X0,…,Xn and Y0,…,Yn ar

binomial queues

› Xi and Yi are of type Bi or null

null// is

carry

//initial null;

:=

C0

do n

to 0 =

i for

ne

and

Zi form

to

Ci and

Xi,Yi, combine

Cn+1:=

Zn+1

Binomial Queues - Lecture 12 10/25/02

2

10 7

12

22 = 4 **N=510=1012**

Binomial Queues - Lecture 12 Exercise

9 4

8

20 = 1 21 = 2

2 = 42

**N=310=112**

10/25/02

O(log N) time to Merg

keys there are at most log2 • For N

trees in a binomial forest.

• Each merge operation only looks root of each tree.

• Total time to merge is O(log N).

Binomial Queues - Lecture 12 10/25/02

• Create a single node queue B0 the new item and merge with

Binomial Queues - Lecture 12 Insert

existing queue

O(log N) time•

10/25/02

1. Assume we have a binomial forest X02. Find tree Xk with the smallest root

4. Remove root of Xk (return this value)

This yields a binomial forest Y0, Y1, …,Yk

5. Merge this new queue with remaindeoriginal (from step 3)

Binomial Queues - Lecture 12 DeleteMin

the queue 3. Remove Xk from

›

Total time = O(log N)•

10/25/02

Implementation

Binomial forest as an array of multiway trees

› FirstChild, Sibling pointers

•

5

4

3

2

1

0

5

2

1

1

2

5

9

10

7

4

7

4

9

12

8

13

1 8

13 15

15

Binomial Queues - Lecture 12 10/25/02

DeleteMin Example

5 4 3 2 1 0

7 6 5 4 3 2 1 0

Rem

2

5

1

2

5

4 9

10 7 4

9

8 13

12 8

13

1

15

15

Binomial Queues - Lecture 12 10/25/02

4

3

2

1

0

Old forest

7

6

5

4

3

2

1

0

2

5

2

5

9

9

4

3

2

1

0

New forest

10

7

4

4

7

10

12

8

13

13 15 12

15

Binomial Queues - Lecture 12 10/25/02

7 6 5 4 3 2 1 0

4 3 2 1 0

2

5

Merge

9

2 5

4

10

7

6

5

4

3

2

1

0

13

4

7

10

15

8

13

12

15

Binomial Queues - Lecture 12 10/25/02

Why Binomial?

*B2*

*B3*

*B4*

)! !

*k*

2 3

4 tree depth *d*

1, 2, 1 1, 3, 3, 1 1, 4, 6, 4, 1 nodes at depth *k*

Binomial Queues - Lecture 12

! *d*

*k d*

10/25/02

*d*

(

*k*

Other Priority Queue

• Leftist Heaps

› O(log N) time for insert, deletemin,

• Skew Heaps

› O(log N) amortized time for insert,

deletemin, merge

• Calendar Queues

› O(1) average time for insert and del

Binomial Queues - Lecture 12 10/25/02

Exercise Solution

1 13 2

9 4

15

10

7

+

8

12

1 2

9

10 7

4

12

8 13

15

Binomial Queues - Lecture 12 10/25/02