Aurelio-Ceejay Guiking

Dr. Wolfgang Bein

CS 302 - 1003

28 April 2018

Assignment 11

1. Quicksort

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | 2 | 5 | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 5 | 3 | 6 | 7 | 4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 3 | 6 | 7 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 6 | 7 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 7 | 6 |

**Sorted:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

1. Heapsort

4 | 2 | 5 | 3 | 6 | 7 | 1

4 | 2 | 5 | 3 | 6 | 1 | 7

4 | 2 | 1 | 3 | 6 | 5 | 7

2 | 1 | 3 | 6 | 5 | 4

Done: 7

2 | 1 | 3 | 4 | 5 | 6

Done : 6, 7

1 | 3 | 4 | 5 | 2

1 | 3 | 4 | 2 | 5

1 | 2 | 4 | 3 | 5

Done : 5, 6, 7

2 | 4 | 3 | 1

2 | 1 | 3 | 4

Done : 4, 5, 6, 7

1 | 3 | 2

1 | 2 | 3

Done : 3, 4, 5, 6, 7

2 | 1

Done : 2 , 3, 4, 5, 6, 7

**Sorted:**

**1 | 2 | 3 | 4 | 5 | 6 | 7**

**Idea: find maximum, take maximum out. Swap the lowest element and put it as the highest element. Compare that element to every single element, until the highest is found. Recursively repeat until every element is taken out.**

1. Mergesort

8 | 4 | 2 | 5 | 3 | 6 | 7 | 1

Split array in half

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 4 | 2 | 5 |  | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 4 | 2 | 5 |  | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 4 |  | 2 | 5 |  | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 8 |  | 2 | 5 |  | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 8 | 2 | 5 |  | 3 | 6 | 7 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 4 | 5 | 8 |  | 3 | 6 |  | 7 | 1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 4 | 5 | 8 |  | 3 | 6 |  | 1 | 7 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 4 | 5 | 8 |  | 1 | 3 | 6 | 7 |

**Sorted:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |

**Idea:**

**Split the table evenly in half, focus on the left side first, split the left in half once more, sort the left side. Repeat the same thing to the right side. Then sort the table starting from the minimum of left and right, repeatedly comparing, until the array is sorted.**

* 1. Bk, height is k and there are exactly 2k nodes

Proof by induction:

**2^(k-1)+ 2^(k-1) = 2^k**

* 1. Bk , exactly (ki) nodes at depth for i = 0

For i = 0

D(k,i) = D(k-1,i) + D(k-1,i-1)

= (k -i 1) + (ki -- 11)

= (ki)

1. Show that
   1. root of binomial tree Bk has degree k

Proof by induction:

B0 has height 0.

Bk-1 has height k - 1.

For Bk, one of the Bk-1’s become the root and hence the height increases by one when the other Bk-1 is attached. Thus, the height of Bk is k - 1 + 1 = k

* 1. Maximum degree of any node in an N-node binomial tree is log2N

No two binomial trees in the collection have the same size.

For all n >= 1 and K >= 0, Bk appears in an n-node binary heap if and only if the (k+1)st bit of the binary representation of n is a 1.

Meaning, that the number of trees in a binomial heap is log2N

O(logn)

1. 0, …., 6 with hash(x) = x mod 7, insert the numbers 1, ….. , 100. How many elements will be at index location 0?

100 / 7 = 14

14 elements will be at index location 0

1. Into a hash table with TableSize = 10, insert the following elements 10, 49, 39, 28, 19 in this order
   1. using linear probing

|  |  |
| --- | --- |
| 0 | 10 |
| 1 | 39 |
| 2 | 19 |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 | 28 |
| 9 | 49 |

* 1. using quadratic probing

|  |  |
| --- | --- |
| 0 | 10 |
| 1 |  |
| 2 |  |
| 3 | 39 |
| 4 |  |
| 5 | 19 |
| 6 |  |
| 7 |  |
| 8 | 28 |
| 9 | 49 |