GIT Department of Computer Engineering

CSE 222/505 - Spring 2021

Homework 2 Report

Muhammed Bedir ULUCAY

1901042697

1. SYSTEM REQUIREMENTS

Program requirements for work properly:

Minimum object size is 16 bytes for modern 64-bit JDK since the object has 12-byte header, padded to a multiple of 8 bytes. In 32-bit JDK, the overhead is 8 bytes, padded to a multiple of 4 bytes. If we accep this a reference size we need to ;

Heap; need 128 byte for class

Nearly 3200 byte for heap test case

Binary Search Heap Tree; need 256 byte for class

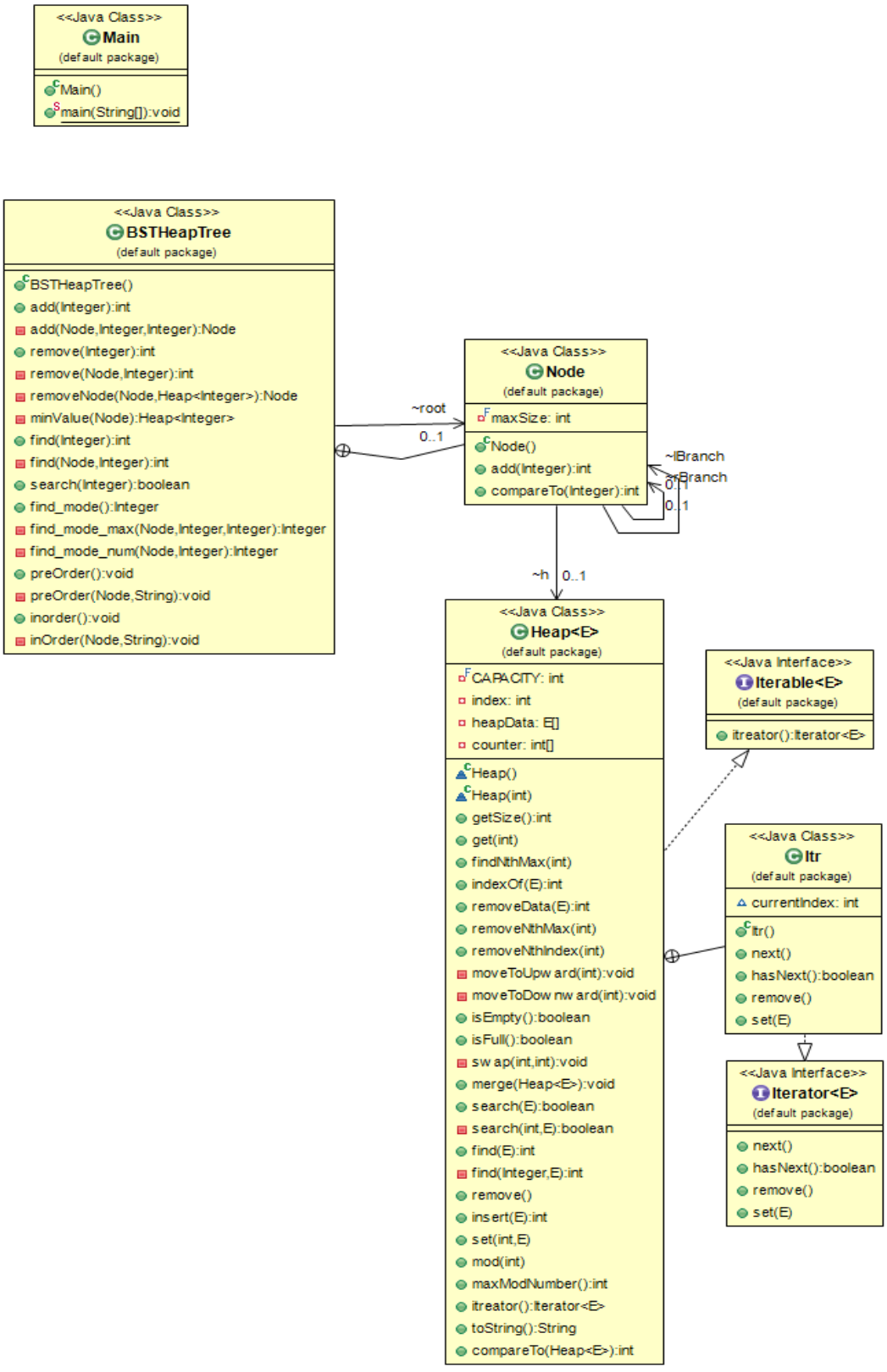
3000 * 16 + 3000 * 8 need byte test case

OS requirements:

Operating System need to jdk and jre for start the program

You need to enough space to store data's of program according to the how many you have objects.

2. CLASS DIAGRAMS



3. PROBLEM SOLUTION APPROACH

Generally problem is adding, removing, searching, replacing and following the structural properties for each structure.

For Heap Structure;

Adding new element placing the end of the list and arrange the max or min heap swap parents until reach the root if it necessary also if necessary we can swap with child of when we want to set etc.

Removing an element also use parent child relation so that if we need to swap child to parent or parent to child we need to swap until follow the rule.

Search for and element we need to look each nood because there is no specific location for any element in the heap we can just known max or min element is root other can change according to the inserting queue.

For Binary Search Heap Tree;

Adding new element is existing two step if the current node heap is not full we adding this heap other wise we compare the root and we choose the direction to insert place for adding element.Until find or making a new heap node in the binary search heap tree.

Removing and element is also has two steps search in current heap node if it is not in we are going to next node according to the comparision if we find we decrease occurrence at once also if the heap is empty we are deleting the binary search heap tree node.

For Searching using normal binary search tree search recursive approach.

4. TEST CASES

Heap Class:

i. Search for an element

ii. Merge with another heap

iii. Removing ith largest element from the Heap

iv. Extend the Iterator class by adding a method to set the value (value passed as parameter) of the last element returned by the next methods.

Binary Search Heap Tree:

i. Insert element.

ii. Search element.

iii. Find the mode of the BSTHeapTree.

iv. Remove element.

5. RUNNING AND RESULTS

Heap Class:

Insertion Element

```
Created heap
[(99,1),(95,1),(97,1),(83,1),(94,1),(61,2),(49,1),(19,2),(34,2),(64,2),(93,1),(58,1),(43,1),(7,1),(32,2),(18,1),(0,1),(2,1),(24,1),(41,1),(14,1),(75,1),(82,1),(9,1),(42,1),(5,1)]
```

Search Element:

```
Search 41 in heap: true
Search 201 in heap: false
```

Removing nth largest:

```
remove 6th largest :82
[(99,1),(95,1),(97,1),(83,1),(94,1),(61,2),(49,1),(19,2),(34,2),(64,2),(93,1),(58,1),(43,1),(7,1),(32,2),(18,1),(0,1),(2,1),(24,1),(41,1),(14,1),(75,1),(5,1),(9,1),(42,1)]
```

Merge List two heap:

```
Merge two list
List1 :[(99,1),(95,1),(97,1),(83,1),(94,1),(61,2),(49,1),(19,2),(34,2),(64,2),(93,1),(58,1),(43,1),(7,1),(32,2),(18,1),
List2 :[(94,1),(91,1),(82,1),(87,1),(88,1),(67,2),(61,1),(65,1),(69,1),(63,1),(57,1),(50,1),(58,1),(28,1),(16,1),(3,2),
```

```
(32,2),(18,1),(0,1),(2,1),(24,1),(41,1),(14,1),(75,1),(5,1),(9,1),(42,1)]
,(16,1),(3,2),(7,2),(12,1),(52,1),(41,1),(1,1),(17,1),(5,1),(18,1),(26,1),(8,1),(39,1),(4,1)]
```

Merged List:

```
Merged List :
[(99,1),(93,1),(97,1),(91,1),(88,1),(67,2),(95,1),(83,1),(87,1),(63,1),(75,1),(50,1),(58,2),(61,2),(94,2),(65,1),(34,1),(69,1),(52,1),(41,2),(24,1),(57,1),(42,1),
(57,1),(42,1),(18,2),(26,1),(8,1),(39,1),(4,1),(28,1),(16,1),(82,1),(3,2),(49,1),(7,3),(19,1),(12,1),(64,1),(43,1),(32,1),(0,1),(2,1),(1,1),(14,1),(17,1),(9,1),(5,2)]
```

Iteration:

Before Iteration:

```
We use iter all list and print it 10 and 15 th element are setting with random value
 in iteration and list print again and of the iteratiron
```

```
Before iteration
[(99,1),(93,1),(97,1),(91,1),(88,1),(67,2),(95,1),(83,1),(87,1),(63,1),(75,1),(50,1),(58,2),(61,2),(94,2),(65,1),(34,1),(69,1),(52,1),(41,2),(24,1),(57,1),(42,1),
```

```
(57,1),(42,1),(18,2),(26,1),(8,1),(39,1),(4,1),(28,1),(16,1),(82,1),(3,2),(49,1),(7,3),(19,1),(12,1),(64,1),(43,1),(32,1),(0,1),(2,1),(1,1),(14,1),(17,1),(9,1),(5,2)]
```

In Iteration:

```
99
93
97
91
88
67
95
83
87
63
75
75 10th are setting random number : 83
50
58
61
94
65
65 15th are setting random number : 28
34
69
52
41
24
57
42
18
26
8
39
4
```

```
39
4
28
16
82
3
28
7
19
12
64
43
32
0
2
1
14
17
9
5
```

After Iteration:

```
After iteration rearranged heap
[(99,1),(93,1),(97,1),(91,1),(88,1),(67,2),(95,1),(83,1),(87,1),(63,1),(83,1),(50,1),(58,2),(61,2),(94,2),(49,1),(34,1),(69,1),(52,1),(41,2),(24,1),(57,1),(42,1),
,(57,1),(42,1),(18,2),(26,1),(8,1),(39,1),(4,1),(28,1),(16,1),(82,1),(3,2),(28,1),(7,3),(19,1),(12,1),(64,1),(43,1),(32,1),(0,1),(2,1),(1,1),(14,1),(17,1),(9,1),(5,2)]
```

Binary Search Heap Tree Class:

Part Of Tree Inserting:

```
h- l - l - l - l - l - r - l - r [(2742,2),(2721,1),(2740,1),(2689,2),(2699,1),(2696,1),(2703,1)]
h- l - l - l - l - l - r - l - r - l [(2733,1),(2716,2),(2730,1),(2688,1),(2701,1),(2710,1),(2723,1)]
h- l - l - l - l - l - r - l - r - l - l [(2732,1),(2718,1),(2686,1),(2712,1),(2714,1),(2683,1),(2685,1)]
h- l - l - l - l - l - r - l - r - l - l - l [(2715,1),(2697,1),(2708,1),(2693,1),(2695,1),(2707,1)]
h- l - l - l - l - l - r - l - r - l - r [(2741,1),(2734,1),(2737,1)]
h- l - l - l - l - l - r - l - r - r [(2758,1),(2752,1),(2753,1),(2745,1),(2751,2),(2748,1),(2749,2)]
h- l - l - l - l - l - r - l - r - r - l [(2755,1)]
h- l - l - l - l - l - r - l - r - r - r [(2760,1)]
h- l - l - l - l - l - r - r [(2887,2),(2860,3),(2865,1),(2775,1),(2778,1),(2799,2),(2861,2)]
h- l - l - l - l - l - r - r - l [(2877,1),(2811,1),(2850,2),(2772,1),(2806,1),(2776,1),(2833,2)]
h- l - l - l - l - l - r - r - l - l [(2859,2),(2787,2),(2800,1),(2780,2),(2785,1),(2795,3),(2793,1)]
h- l - l - l - l - l - r - r - l - l - l [(2844,1),(2830,1),(2834,1),(2766,3),(2804,1),(2797,2),(2792,1)]
h- l - l - l - l - l - r - r - l - l - l - l [(2839,1),(2838,1),(2827,1),(2773,1),(2828,2),(2826,1),(2783,1)]
h- l - l - l - l - l - r - r - l - l - l - l - l [(2825,1),(2823,1),(2819,1),(2794,1),(2767,1),(2764,1),(2809,1)]
h- l - l - l - l - l - r - r - l - l - l - l - r [(2841,1)]
h- l - l - l - l - l - r - r - l - l - l - r [(2854,2),(2851,1),(2846,1),(2845,1)]
h- l - l - l - l - l - r - r - l - l - r [(2875,2),(2869,1),(2871,1),(2862,1),(2863,1),(2866,1),(2868,1)]
h- l - l - l - l - l - r - r - l - r [(2881,1),(2878,2)]
h- l - l - l - l - l - r - r - r [(2902,1),(2895,1),(2898,1),(2890,1),(2894,1),(2896,1),(2888,1)]
h- l - l - l - l - l - r - r - r - l [(2897,1),(2891,1)]
h- l - l - l - l - l - r [(2943,1),(2926,3),(2938,1),(2918,1),(2906,2),(2912,1),(2923,1)]
h- l - l - l - l - r - l [(2940,1),(2930,2),(2935,1),(2909,1),(2911,1),(2934,1),(2932,1)]
h- l - l - l - l - r - l - l [(2937,1),(2920,1),(2907,3),(2917,1)]
h- l - l - l - l - r - l - r [(2942,1),(2941,1)]
h- l - l - l - l - r - r [(2950,2)]
h- l - l - l - r [(3105,1),(3021,2),(3050,1),(3008,1),(3000,1),(3032,2),(3037,3)]
h- l - l - l - r - l [(3097,1),(2985,2),(3060,1),(2966,2),(2971,2),(3005,1),(3054,2)]
h- l - l - l - r - l - l [(3090,1),(3064,1),(3041,1),(3009,2),(2964,1),(2973,2),(3023,1)]
h- l - l - l - r - l - l - l [(3083,2),(3065,1),(3074,1),(3019,1),(3039,1),(2978,1),(3066,2)]
h- l - l - l - r - l - l - l - l [(3027,1),(3013,1),(2998,1),(2960,1),(3006,2),(2958,1),(2957,1)]
h- l - l - l - r - l - l - l - l - l [(3011,2),(2995,1),(3002,1),(2954,1),(2981,1),(2977,2),(2986,1)]
h- l - l - l - r - l - l - l - l - l - l [(2970,1)]
h- l - l - l - r - l - l - l - l - l - r [(3018,1)]
```

Occurance of an element:

```
1735 occurance = 2
1738 occurance = 2
1738 occurance = 2
1740 occurance = 2
1740 occurance = 2
1741 occurance = 1
1742 occurance = 4
1742 occurance = 4
1742 occurance = 4
1742 occurance = 4
1743 occurance = 1
1745 occurance = 2
1745 occurance = 2
1747 occurance = 1
1748 occurance = 1
1750 occurance = 1
1751 occurance = 1
```

Search:

```
148 search = true
149 search = true
151 search = true
157 search = true
158 search = true
160 search = true
161 search = true
162 search = true
165 search = true
166 search = true
166 search = true
```

```
5001 search = false
5050 search = false
-58 search = false
-190 search = false
9999 search = false
6542 search = false
-654 search = false
-6156 search = false
```

Finding Mode:

```
====================MODE====================


find_mode() = 1742
find(find_mode()) = 4
```

Remove:

```
================REMOVE================


in List remove
29 = 0
47 = 1
69 = 0
94 = 0
123 = 0
142 = 1
166 = 1
203 = 0
226 = 1
254 = 0
282 = 1
318 = 3
335 = 0
354 = 1
```

```
not in List remove
-1 = -1
-2 = -1
5001 = -1
5050 = -1
-58 = -1
-190 = -1
9999 = -1
6542 = -1
-654 = -1
-6156 = -1
```