

2018-03-29 HW #2 and Exam #2 Review

Thursday, March 29, 2018 8:58 AM

CS 211 Homework #2

- make sure to practice your knowledge using an online tool such as Visualization homepage: <http://www.cs.usfca.edu/~galles/visualization/Algorithms.html>

1. [3] Remove 5 from the following AVL tree; draw the results:

| Original | After removal |
|--------------------|---|
| | |
| After 1st Rotation | After 2st Rotation (may or may not be required) |
| | |

2. [3] Insert the value "8" into the following AVL tree. Draw the result:

| Original | After removal |
|----------|---------------|
| | |

| After 1st Rotation | After 2st Rotation (may or may not be required) |
|--------------------|---|
| | |

3. [3] Add 45 to the following AVL tree.

| Original | After insert |
|--------------------|---|
| | |
| After 1st Rotation | After 2st Rotation (may or may not be required) |
| | |

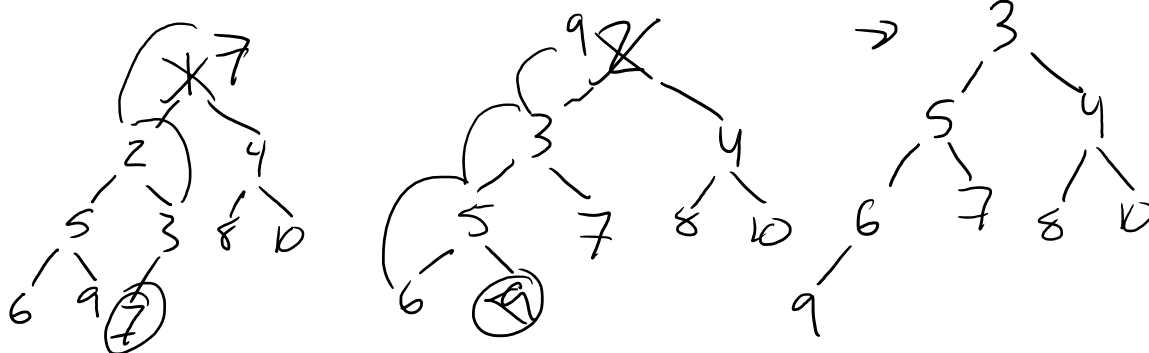
4. **Binary Heaps** Starting with an empty binary **min heap**, show the following. Be sure to clearly label each diagram

4. **Binary Heaps** Starting with an empty binary min heap, show the following. Be sure to clearly label each diagram.

- A. [3] The final state of the heap, in tree form, after adding in the values: 5, 4, 3, 6, 7, 8, 10, 2, 9, 1



- B. [2] The state of the heap, in tree form, after two Dequeue() operations

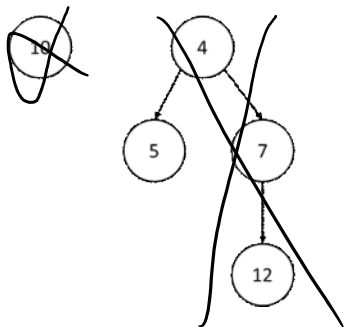


- C. [1] The final, array-based version of the heap

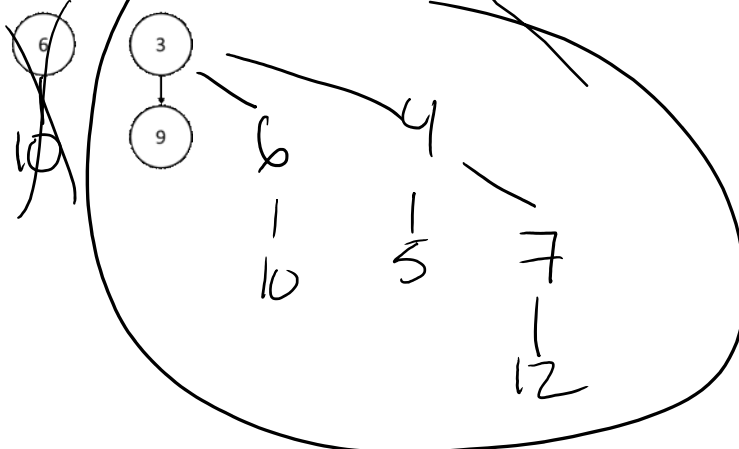
| | | | | | | | | | |
|---|---|---|---|---|---|----|---|---|---|
| 3 | 5 | 4 | 6 | 7 | 8 | 10 | 9 | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

5. [3] Merge the following two binomial heaps

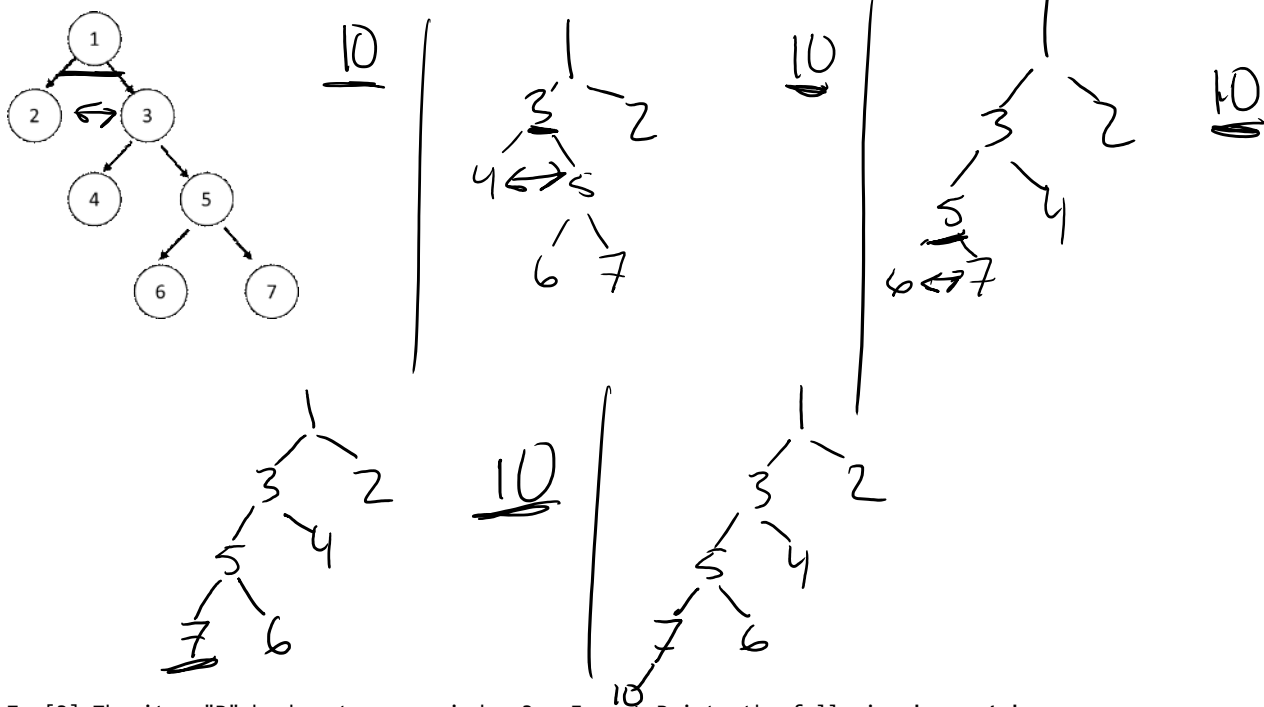
Binomial Heap #1



Binomial Heap #2



6. [3] Enqueue 10 to the following skew heap. Draw the results.

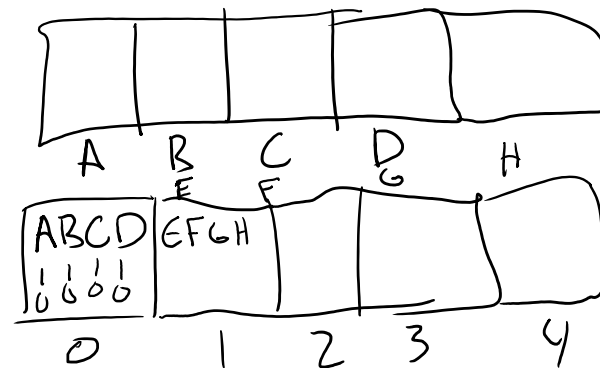


7. [3] The item "B" hashes to array index 3. Insert B into the following hopscotch hashtable whose max distance is 4.

| A | C | E | G | D | H | I | | | |
|------|------|------|------|------|------|------|---|---|---|
| 1100 | 0100 | 0010 | 1000 | 0000 | 1100 | 0000 | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

RESULT:

| A | C | E | G | D | H | B | I | | |
|------|------|------|------|------|------|------|------|---|---|
| 1100 | 0100 | 0010 | 1001 | 0000 | 1010 | 0000 | 0000 | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |



Exam Review

- More algorithm analysis questions
 - How many times will the loop execute, give efficiency (see exam #1)
 - Conceptual AA question related to improving performance
 - E.g. What causes this to run slow? How this be improved? What is the likelihood of this happening?
- Draw a binary tree given pre/in/post traversals
- AVL rotation questions (see HW #2)
- Priority queue questions (min heap; see HW #2)
 - Binary heap
 - Binomial heap
 - Skew heap
- Hash table questions (see HW #2, additional examples below)
 - Where will this number be placed in the HT (open addressing hash table)

- Conceptual hash table questions
 - What affects HT performance
 - How do hand deletes
 - Explain what happens on a resize
 - Etc.
- More programming questions
 - Related to recursive tree programming
 - Questions that appear difficult unless you use a specific data structure
 - Might be trees, HTs, priority queues, etc.
 - Example: Determine if two BSTs are exactly the same

More Examples

Cuckoo Hash Table

| | | | | | |
|-----|-----|----------------|---|---|----------------|
| B E | A F | D H | | | F G |
| 0 | 1 | 2 | 3 | 4 | 5 |
| D | C | B | A | | |

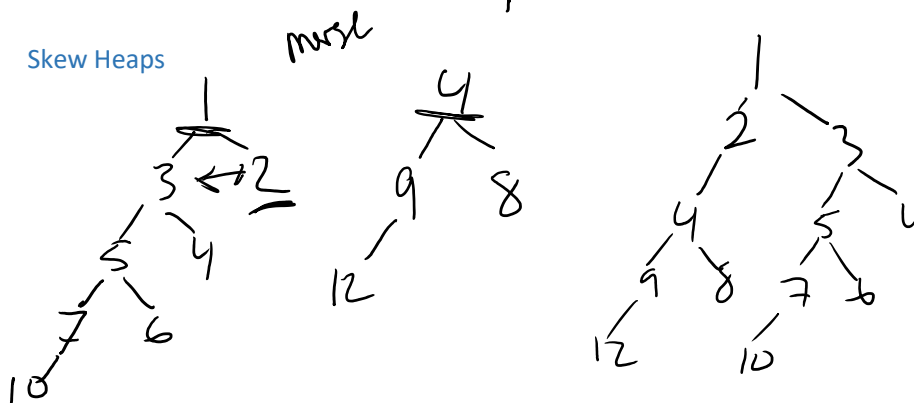
↓
 A: 1, 3
 B: 0, 2
 C: 5, 1
 D: 2, 0
 E: 0, 0
 F: 1, 1
 G: 5, 3
 H: 2, 2

Hopscotch Hashing

A: 1, B: 0, C: 5, D: 2, E: 0, F: 1, G: 5, H: 2

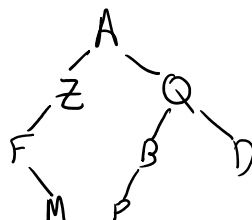
| | | | | | | | | | |
|------|------|------|----------------------------|------|----------------------------|----------------|----------------|----------------|---|
| B | A | D | E | F | H | G | C | | |
| 0 | 1 | 2 | 3 | 4 | 5 _A | 6 _B | 7 _C | 8 _D | 9 |
| 1001 | 1001 | 1001 | ^{A B C D} 0000 | 0000 | ^{A B C D} 0110 | 0000 | 0000 | | |

Skew Heaps



Draw A tree

← →
 IN: F, M, Z, A, P, B, Q, D
 Post: M, F, Z, P, B, D, Q, A
 1 1 1 1 1 1 1



Binomial Heap

