

HW 5

1. Assume a sorted link list is used to implement a min priority queue that holds integer objects. Show the contents of the link list after the following sequence of insert and deleteMin operations. Separate your entries of the link list with commas; e.g. "3, 82, 1" but do not put in the double quotes.

insert(14); insert(12); insert(29); insert(16); insert(5); insert(7); deleteMin();

A. I will show all the steps

14

12,14

12,14,29

12,14,16,29

5,12,14,16,29

5,7,12,14,16,29

deleteMin();

7,12,14,16,29

B. Assume a sorted link list is used to implement a min priority queue that holds integer objects. What is the worst-case runtime for the insert operation and why? The worst-case would-be $O(n)$ because you might have to iterate through the whole linked list to insert the value at the end of the list.

C. Assume a sorted link list is used to implement a min priority queue that holds integer objects. What is the worst-case runtime for the deleteMin operation and why? The worst case would be $O(1)$ because the linked list is sorted from least to greatest. Therefore the minimum would always be the first element in the list.

2. Assume a non-sorted link list is used to implement a min priority queue that holds integer objects. Show the contents of the link list after the following sequence of insert and deleteMin operations. Separate your entries of the link list with commas; e.g. "3, 82, 1" but do not put in the double quotes.

insert(14); insert(12); insert(29); insert(16); insert(5); insert(7); deleteMin();

A. I will show all steps

14

12, 14

29, 12, 14

16, 29, 12, 14

5, 16, 29, 12, 14

7, 5, 16, 29, 12, 14

deleteMin();

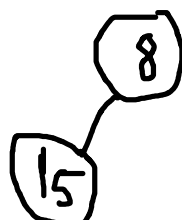
7, 16, 29, 12, 14

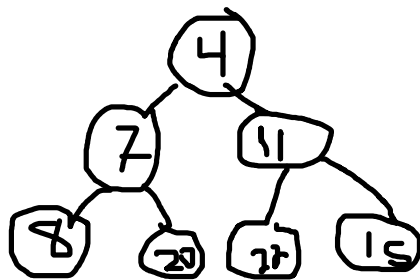
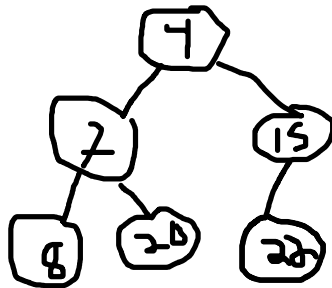
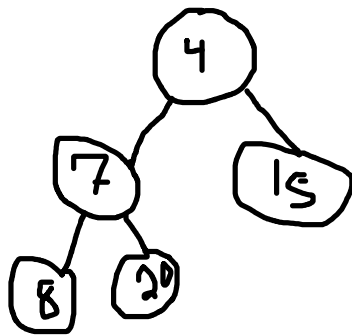
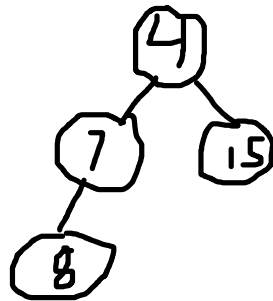
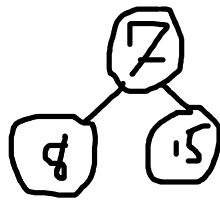
B. Assume a non-sorted link list is used to implement a min priority queue that holds integer objects. What is the worst-case runtime for the insert operation and why? The worst-case runtime for the insertion operator would be $O(1)$ because when inserting into a nonsorted linked list it will just insert to the front of the list and push every other item back

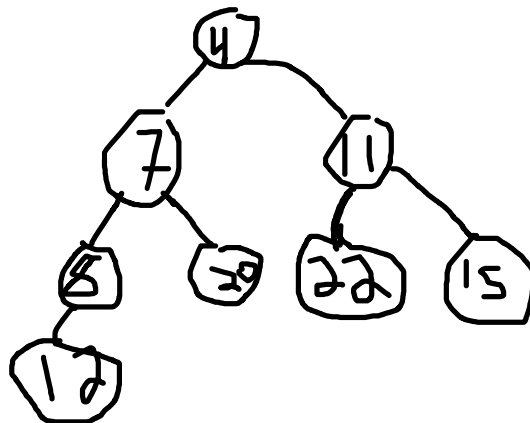
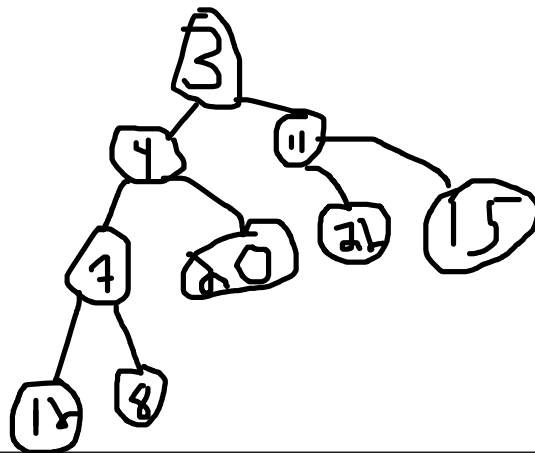
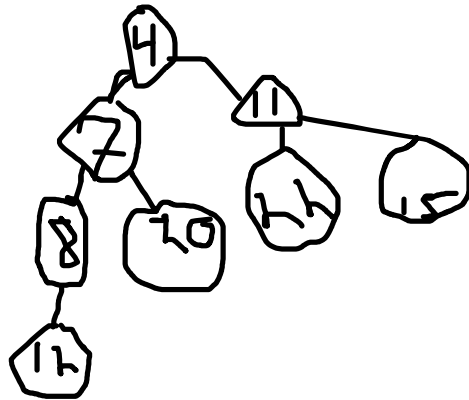
C. Assume a non-sorted link list is used to implement a min priority queue that holds integer objects. What is the worst-case runtime for the deleteMin operation and why? The worst-case runtime for the deletedMin functionality of the unsorted linked list would take $O(n)$ or constant time because you may have to search through the entire list to delete the minimum value.

3. Per our course book and lectures, assume a min binary heap is used to implement a priority queue that holds integer objects. Show the contents of the underlying array after the following insert

A. I will show all my steps yet again







Null, 4, 7, 11, 8, 20, 22, 15, 12

- B. Assume a min binary heap is used to implement a priority queue that holds integer objects. What is the worst case runtime for the deleteMin operation and why?

The worst-case runtime would be $O(\log n)$ because the deleteMin because you would have to reorder the heap by rearranging one of the sides.

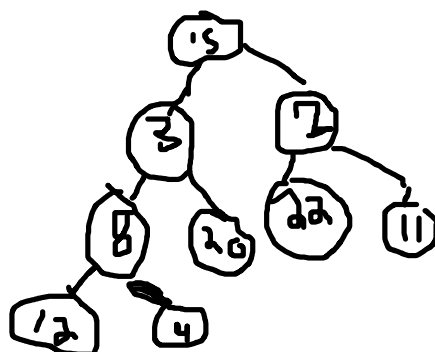
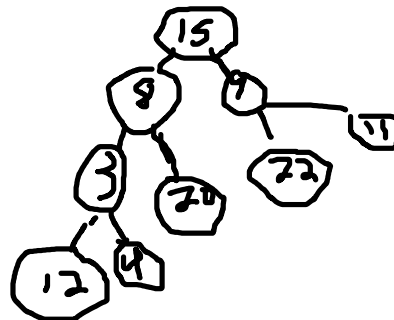
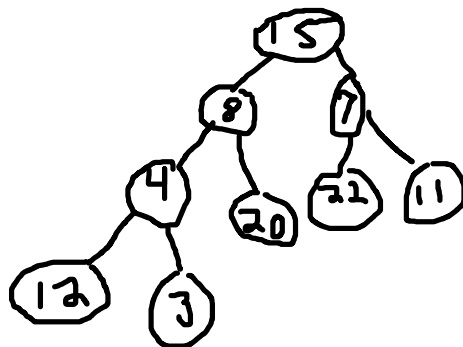
- C. Assume a min binary heap is used to implement a priority queue that holds integer objects. What is the worst case runtime for the insert operation and why?

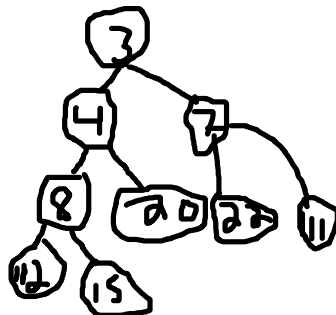
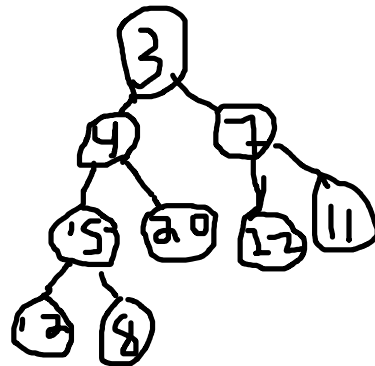
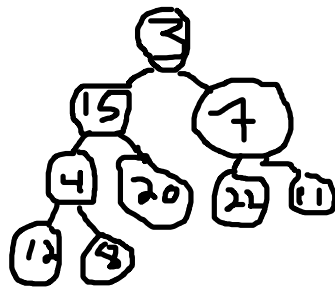
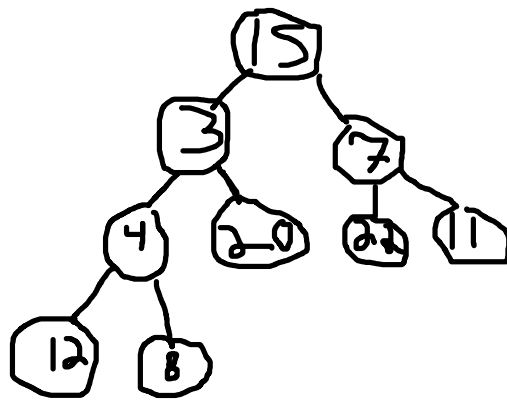
The worst case runtime for the insertion operation would also be $O(\log n)$ because you may have to search all the way down one side of the heap to insert a value

Per our course book and lectures, assume a min binary heap is used to implement a priority queue that holds integer objects. Assume the min binary heap is instantiated with the vector that contains the following sequence of integers:

15, 8, 7, 4, 20, 22, 11, 12, 3

Show the contents of the underlying array after the binary heap is initialized. Separate your entries with commas; e.g. "3, NULL, 1" but do not put in the double quotes. Aside: the constructor is invoked with the vector and the build_heap heapify operation is performed.





Noah Krill
Data Structures
7 April 2020

What is the runtime for building a heap with an initialization list and what is the runtime for building a heap with repeated insert operations? Make sure you state initialization list runtime is... and repeated insert runtime is...

The initialization list run time is $O(n)$.

Insertion into an already existing min heap by repeated insert would be $O(\log n)$.

The C++ STL priority queue container adaptor that is used to implement a heap. Does it implement a min heap or a max heap and what is its default container class? Hint: Go to cplusplus.com and read about the priority queue.

The standard container of vector is used. The standard template library uses max heap.

The C++ STL priority queue container adaptor that is used to implement a heap. What property is required of its iterator? Considering this requirement of the iterator, can a Binary Search Tree be used as the container adaptor to implement a C++ STL priority queue, explain why/why not? Hint: Go to cplusplus.com and read about the priority queue.

Well the priority queue uses random access iterators so no a binary search tree cannot be a container for a STL priority queue because they have more rules about having the lesser values be stored on the right and the greater values be stored on the left. Random access iterators would be less efficient at that.