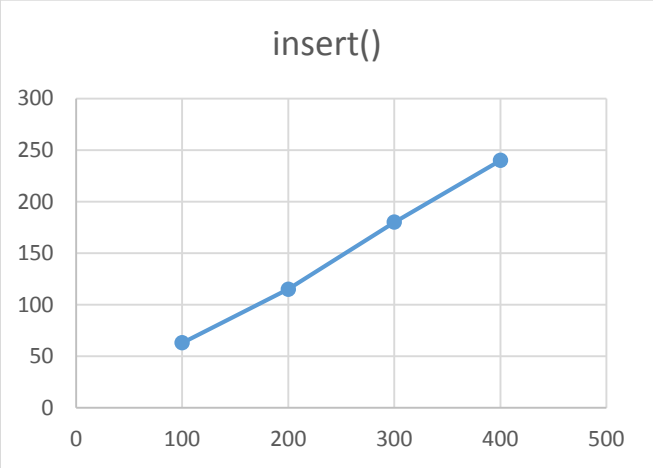
Conner Thomas
10/25/13
CSE373

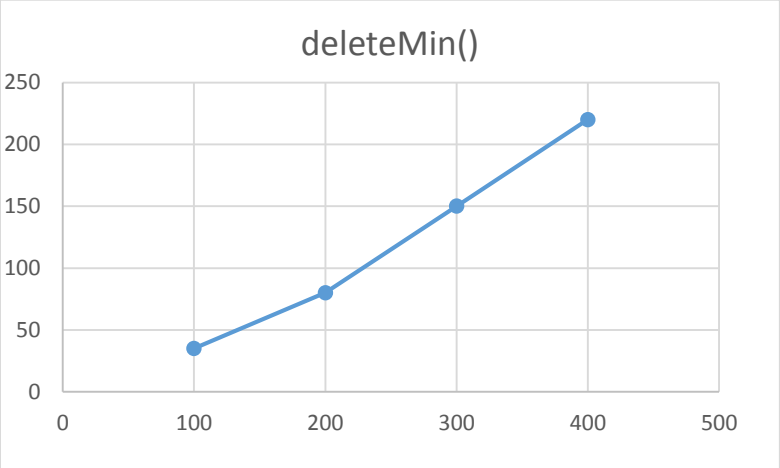
Homework 3 Write Up

1. Runtime predictions

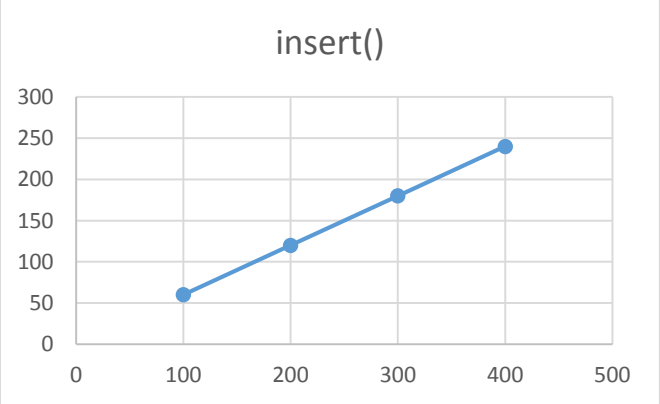
	Binary Heap	ThreeHeap	myPQ
isEmpty	O(1)	O(1)	O(1)
size	O(1)	O(1)	O(1)
insert	O(logn)	O(logn)	O(n)
findMin	O(1)	O(1)	O(1)
deleteMin	O(logn)	O(logn)	O(n)
makeEmpty	O(1)	O(1)	O(1)

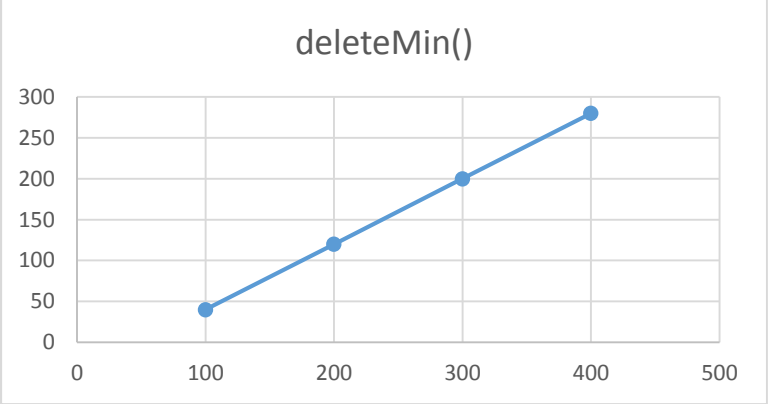
2. Binary Heap timings: (Y axis is time in ms, x axis is n value)



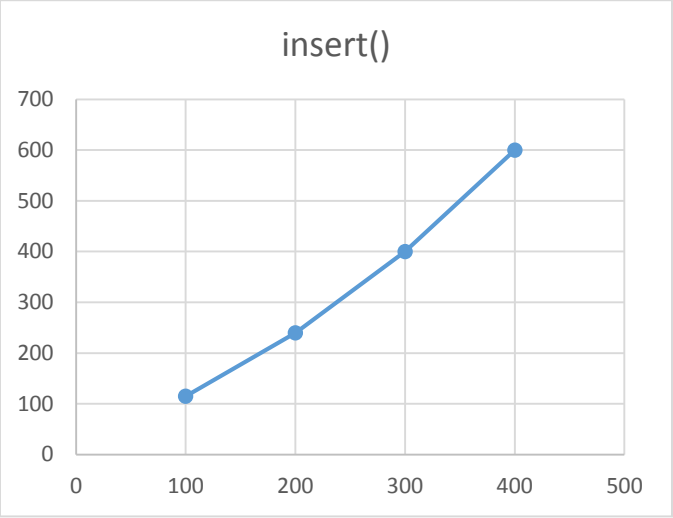


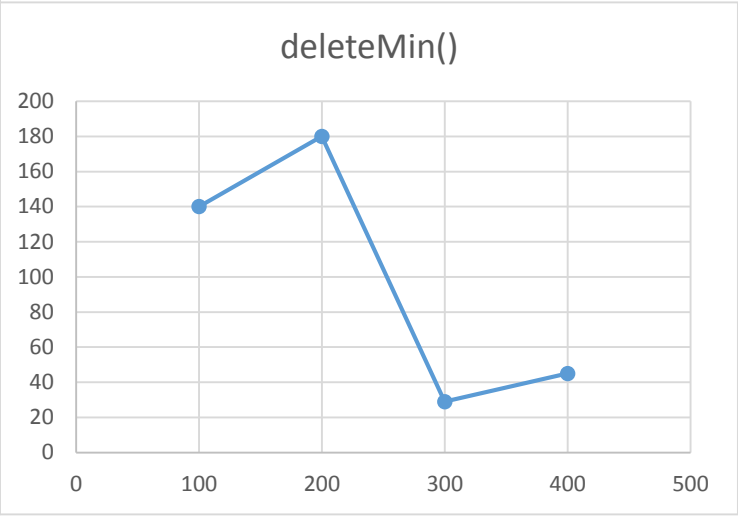
ThreeHeap:





MyPQ:





3a. My analysis was useful and the predictions were somewhat similar to the plotted data that I gathered using my timing program. Insert for both the 2 and 3 heaps seem to resemble a logarithmic fit of data points. DeleteMin does as well for the 3 heap, but doesn't seem to for the 2 heap. I was also nearly correct with the behavior of MyPQ as far as timings go. Insert is on average, O(n) at worst. I'm not exactly sure why deleteMin behaves so strangely in regard to timing, but it is a replicable result.

3b. My prediction was very far off for deleteMin for the MyPQ. I'm not sure if it follows a pattern as n grows. The drop in time is replicable but I'm not ready to write it off as a quirk of Eclipse or some other aspect of the computer I'm working on. If the processor is not solely working on running the code, then it could get interrupted while it is trying to do so, which could lead to some timing issues.

3c. I would recommend to anyone who needs to use a heap to just use a binary heap. It has been very thoroughly tested in industry and has a good balance for being able to search very quickly yet also being able to rearrange without too much computation. I cannot think of a case where I'd suggest anything else.

4. I tested my heap implementations using a class that used for loops and math.random to fill the heaps with random values between 1 and 2. I would then pull away all of the values and make sure that the heap new that it was empty. I also used the same class to ensure that makeEmpty(), size(), and all the other non-algorithmic methods worked using the debugger and some specific numbers that would fill then empty a heap, then check if it was empty.

5a.

	Binary heap	3-heap	4-heap	5-heap
Placement of child nodes of node i in array	i*2 i*2 + 1	i*3 - 1 i*3 i*3 + 1	i*4 – 2 i*4 – 1 i*4 i*4 + 1	i*5 – 2 i*5 – 1 i*5 i*5 + 1 i*5 + 2

5b. Formula for left-most node in row i:

Position = (i * d) – i/2