Minh Nguyen

7/20/2014

Problem 2 PDF

| n | 1000 | 2000 | 4000 | 8000 | 16000 | 32000 | 64000 | 128000 | 256000 |
|---|---|---|---|---|---|---|---|---|---|
| Dynamic | 0ms | 0ms | 0ms | 0ms | 0ms | 0ms | 0ms | 0ms | 0ms |
| Linklist | 0ms | 0ms | 30ms | 120ms | 510ms / 24kb | 2070ms /524kb | 8260ms /1524KB | 32,340ms /3524KB | 205280ms /7524 |

I was unable to find a way to plot a graph using microsoft word, So I listed my timing tests in a table instead with some very interesting results.

1) Which of the implementations uses more memory? Explain Why.

According to my findings, The linked list used much more memory. I would say this is the case because since linked lists do not have random access to it's data, it must iterate through the entire list to find the right number for the contains() function.

2) Which of the implementations is the fastest? Why?

The Dynamic appeared to be much faster. I checked to see when the Dynamic array actually started showing something other than 0ms, and it was actually at the 512,000 element mark that it showed that it took 10ms for it to run. I would say it's faster because of the binary search function that's possible in the dynamic array, which has a algorithmic complexity of O(log n).

3) Would you expect anything to change if the loop performed removed() instead of contains()? If so, what?

Yes, if it was removed, it would cause the dynamic array to be much slower. As we have looked at before, removing from an array will cause the array to shift down one, which is a O(n) operation, and would drastically slow it down, as opposed to the contains() O(log n) complexity.