CSCI 430: Program2: MaxSubArray

John E. Buckley

October 11, 2017

Implementation:
The implementation for MaxSubArray was fairly straightforward from the book, however I ran into several problems using python, more specifically the latest version of python. It is nice to use things like float("-inf") which only exist on the latest version of python however I was unaware that division now gave back decimals. For example in Python 2: 7/3 was simply 2 but in Python3: 7/3 is 2.3. This caused my program to crash with the error: Maximum recursion depth reached, however this was a quick fix once I realized what the problem was and quite simply I should have done floor division anyway as that was how we implemented it in class. I've also chosen to increase both the Low and High by 1 for convenience.

Testing Expectations:
I expected this to perform almost exactly like merge-sort as the recurrence for MaxSubArray is the same recurrence as merge-sort which is $\Theta(nlgn)$. The master method described in the book shows this.

Testing Observations and Analysis
The results are similar to that of merge-sort, however the random data always runs higher than ordered or reverse-ordered data. I failed to account for how MaxSubArray will react to monotonically increasing and decreasing data.

Take-Away:
Out of the several different ways discussed in class on how to approach the MaxSubArray problem, the best way to implement this is to use Divide and Conquer rather than brute-forcing it. I feel that this will not be the last time I see Divide and Conquer as it tends to be the better way to implement several searching problems, as far as I have seen up to this point.