Largest Subarray

Joseph Watts

February 12, 2019

Abstract

1 Background and Related Work

1.1 Brute Force Algorithm

Algorithm 1 Insertion Sort

```
function InsertionSort(L)
    for i 1..len(L) do
        $j \leftarrow i$
        while $j > 0$ and $L[j] < L[j-1]$ do swap(L[j], L[j-1])
            $j \leftarrow j - 1$
        end while
    end for
end function
```

Explain the function here:

Summation Equation goes here

Explain runtime complexity:

1.2 Kadane Algorithm

Algorithm 2 Merge Sort

```
function MergeSort(L)
    if $len(L) <= 1$ then
        return L
    end if
    $A \leftarrow$MergeSort(first half of L)
    $B \leftarrow$MergeSort(second half of L)

    return Merge(A,B)
end function
```

Explain Kadane Here

Gimme big O calculation pls

Explain big O and big Θ here

2 Experimental Setup

RUST HAS BIG PP OWO

This is how we timed our setup

3 Results

Include our graph visualization of our data here. Brute Force and Kadane Timing

4 Conclusions