Jeonghyeon Woo

CPT 223

HW 4

Feb. 20, 2018

Report)

Problem statement:

By running four different algorithms of maxsubsum, observe which algorithm has better time complexity.

Experimental Setup:

Machine: Samsang laptop

Ram: 8GB, 64bit,

CPU: i5-3230M 2.6GHz

OS: Window 10

compiled by visual studio

number of trial for each size: 10

Result:

(See excel plot) \* y=time x=size

The result agrees with my expectation.

The first algorithm has 3 loops, so maxsubsum1 = O(n^3).

The second algorithm has 2 loops, so maxsubsum2 = O(n^2).

The third algorithm divides the array into two parts and process the parts separately, which means maxsubsum3 = 2nlog(n), so maxsubsum3 = O(nlog(n)).

The fourth algorithm has one loop, so it processes only n times: maxsubsum4 = O(n).

The elapsed times of each algorithms are not so different when n is small. However, the difference become bigger as n becomes bigger.

It was faster in this order: maxsubsum4, maxsubsum3, maxsubsum2, maxsubsum1. The results agree with the algorithms’ time complexity.