Name: Kyle Salitrik
ID: 997543474
Due: 7/27/2013 11:55PM
Last Modified: 7/27/2013 7:48 PM

CMPSC 122 – Project 8: Algorithm Analysis

a) project08algorithm1.cpp
    I. The inner loop runs a total of 5 commands N times, and the outer loop runs 3N times excluding the inner loop. When the algorithms are run together there is a total of cN^2 runs where c is a constant coefficient dropped in Big-O notation. The Big-O notation I obtained was: $O(N^2)$. This algorithm is in the middle of the given algorithms for calculation time.
b) project08algorithm2cpp
    I. The innermost loop runs a set of statements $\frac{N(N+1)}{2}$ times, and the two outermost loops run sets of statements n times each. Putting all of the information together, $\frac{N(N+1)}{2} * N * N$ gives the Big-O notation of: $O(N^4)$ meaning that this algorithm will take the longest out of the three given and shouldn't be used.
c) project08algorithm3.cpp
    I. The worst case scenario of the inner else-if statements is to run n times for either statement. The for loop runs a total of 2N times and the inner statements also run a total of 2N times. This gives a total $T(N) = 2N + 2N + 1 = 4N + 1$ and results in a Big-O notation of $O(N)$, which is the fastest of the given algorithms.
d) Machine Specifications
    I. Lenovo Ideapad Y580

    System Information

    Time of this report: 7/27/2013, 18:25:19
    Machine name: KYLE-LAPTOP
    Operating System: Windows 8 64-bit (6.2, Build 9200) (9200.win8_gdr.130410-1505)
    Language: English (Regional Setting: English)
    System Manufacturer: LENOVO
    System Model: 20132
    BIOS: 5DCN90WW(V8.01)
    Processor: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz (8 CPUs), ~2.4GHz
    Memory: 8192MB RAM
    Available OS Memory: 8058MB RAM
    Page File: 3128MB used, 9536MB available
    DirectX Version: DirectX 11
    DxDiag Version: 6.02.9200.16384 64bit Unicode

e) BLUE (Algorithm 1) Output

I. Maximum contiguous subsequence sum: 105
   Process Timer

   Elapsed Time : 0.001s

II. Maximum contiguous subsequence sum: 235
   Process Timer

   Elapsed Time : 0.001s

III. Maximum contiguous subsequence sum: 457
   Process Timer

   Elapsed Time : 0.001s

IV. Maximum contiguous subsequence sum: 669
   Process Timer

   Elapsed Time : 0.001s

V. Maximum contiguous subsequence sum: 925
   Process Timer

   Elapsed Time : 0.001s

VI. Maximum contiguous subsequence sum: 1260
   Process Timer

   Elapsed Time : 0.007s

VII. Average Time: 0.002s

f) GREEN (Algorithm 2) Output

I. Maximum contiguous subsequence sum: 501
   Process Timer

   Elapsed Time : 0.001s
II. Maximum contiguous subsequence sum: 385
   Process Timer

   Elapsed Time : 0.002s
III. Maximum contiguous subsequence sum: 567
   Process Timer

   Elapsed Time : 0.009s
IV. Maximum contiguous subsequence sum: 697
   Process Timer

   Elapsed Time : 0.057s
V. Maximum contiguous subsequence sum: 603
   Process Timer

   Elapsed Time : 0.433s
VI. Maximum contiguous subsequence sum: 1067
   Process Timer

   Elapsed Time : 3.443s
VII. Average Time: 0.6575s

g) RED (Algorithm 3) Output
   I. Maximum contiguous subsequence sum: 234
      Process Timer

      Elapsed Time : 0.001s
   II. Maximum contiguous subsequence sum: 529
      Process Timer

      Elapsed Time : 0.001s
   III. Maximum contiguous subsequence sum: 529
      Process Timer

      Elapsed Time : 0.001s
   IV. Maximum contiguous subsequence sum: 812
      Process Timer

      Elapsed Time : 0.001s
   V. Maximum contiguous subsequence sum: 812
      Process Timer

      Elapsed Time : 0.001s
   VI. Maximum contiguous subsequence sum: 1025
      Process Timer

      Elapsed Time : 0.001s
   VII. Average Time: 0.001s
h) Summary:
   I. BLUE ALGORITHM (project08algorithm1.cpp): Avg time:0.002s
   II. GREEN ALGORITHM (project08algorithm2.cpp): Avg time: 0.6575s
   III. RED ALGORITHM (project08algorithm3.cpp): Avg time: 0.001s
   IV. The conclusions from running the three algorithms is – in fact – that the third algorithm is the fastest, the first is in the middle and the second algorithm will take the longest out of any of them to compute as was predicted by the Big-O notations. In algorithm 2 vs algorithm 3, if n is greater than some value between 64 and 128, then algorithm 3 will take longer. However comparing algorithm 1 to algorithm 3 shows that when n is greater than some value between 1024 and 2048 then algorithm 3 will be quicker to compute.