Due Monday 11/15/21 **© 2021 Indianapolis, IN**

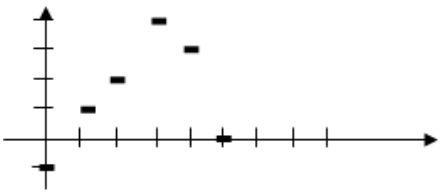
CSCI 48400 Divide-and-Conquer Homework

1. **(10 points)** Use Strassen's algorithm (by hand) – ***use the formulas and notation as shown in the notes*** - to compute **A** · **B** where

$$A = \begin{bmatrix} 2 & 3 & 1 & -2 \\ 5 & 3 & 6 & -1 \\ 1 & 3 & 2 & 2 \\ 0 & 4 & 1 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & 3 & 2 & 1 \\ 4 & 2 & -3 & 1 \\ 1 & 0 & 3 & 2 \\ 2 & -2 & 1 & 4 \end{bmatrix}$$

Show your work in detail – yes, this will be extremely tedious. (Careful bookkeeping is the key. Don't use any "shortcuts" on the 2 × 2 multiplications, do them by Strassen's algorithm as well. And remember that you have a check on the correct matrix results as you go along by just doing ordinary matrix multiplication.) Compute the total number of additions and the total number of multiplications of array elements (numbers) done in this process. Compare these results with the exact expressions we derived for multiplications and additions using Strassen's algorithm. Also compare these results with the work to use standard matrix multiplication. Turn all this in on Canvas via a pdf file **(for the Instructor)**.

2. **(10 points)** Consider the following problem. You have an array A[0], A[1], …, A[n - 1] of distinct integers that has the following property: The values in the array increase up to index p for some p, $0 \le p \le n-1$, and then decrease for all indices beyond p through position n - 1. You want to find the p index at which the peak value occurs and what the peak value is. Example: In the 6-element array illustrated below, the p index is 3, and the peak value is 4.



Describe [write an English paragraph, not code] a divide-and-conquer algorithm that can solve this problem. (Hint: the solution is a bit like the binary search algorithm.) Do a formal analysis (for which you can assume that n is a power of 2) – this means formulate and solve a recurrence relation – to prove that your algorithm does Θ(lg n) work units, where the work unit is comparison of array values. Turn your algorithm description and analysis in on Canvas as a pdf file **(for the TA)**.

Implement your algorithm in a C++ program called *peak.cpp*. Recall your program will be run on Visual Studio 2019. If you use Visual Studio to create your program, ***be sure that you choose Empty Project***. I expect that, on all coding assignments, you will follow good coding practice – reasonable identifier names, modularization using functions, clear comments, etc.

Read the input values (i.e., the array values) as a series of integers, one per line, from a text file called *peak.txt* – make up your own file for testing your program (see the reminder on the C++ Resources page about reading data from a file). You can assume a maximum array size of 32*. Or you can implement using a vector instead of an array; with a vector you don't have to worry about size, the system automatically gives you enough space. Turn in only *peak.cpp* (not all your project files or data files) via Canvas Assignments.

**Note that in a "regular" C++ array, the array size must be declared as a constant value. Read Arrays vs. Vectors on the C++ Resources page.*