AMATH 483 / 583 (Roche) - Homework Set 5

Due Friday May 9, 5pm PT

May 2, 2025

Homework 5 (80 points)

1. (+20) Ring. $\forall x, y \in \mathbb{Z}$ define binary operator $\oplus$ as $x \oplus y = x + y - 1$, and binary operator $\odot$ as $x \odot y = x + y - xy$. $(\mathbb{Z}, \oplus, \odot)$ is a ring.
   (a) Additive Identity. Find $z \in \mathbb{Z} \ni a \oplus z = z \oplus a = a, \forall a \in \mathbb{Z}$.
   (b) Additive Inverse. $\forall a \in \mathbb{Z}, \exists b \in \mathbb{Z} \ni a \oplus b = b \oplus a = z$. Find b.
   (c) Show that $\oplus$ commutes.
   (d) Multiplicative Identity. A ring with unity is a ring with member $u \in \mathbb{Z}, u \neq z \ni a \odot u = u \odot a = a, \forall a \in \mathbb{Z}$. Find u.
2. (+10) Recursive sequence. Let $T_0 = 2$, $T_1 = 3$, and $T_2 = 6$, and for $n \geq 3$ let $T_n = (n+4)T_{n-1} - 4nT_{n-2} + (4n-8)T_{n-3}$.
   (a) Derive a formula of the form $T_n = A_n + B_n$ where $(A_n), (B_n)$ are well-known sequences.
   (b) Use induction to prove your formula works.
3. (+20) Memory access time. Write C++ functions that perform **row and column swap** operations on a type double matrix stored in column major index order using a single vector container for the data. The specifications are provided here. Put the functions in file **matrix_swaps.hpp** (no need for header guards -just the code for your functions). Conduct a performance test for square matrix dimensions 16, 32, 64, ... 4096, measuring the time required to conduct row swaps and column swaps separately. Let each operation be measured *ntrial* times, *ntrial* ≥ 3. Make a single plot of your *row* and *column* swap timing measurements with time on the y-axis (could be $log_{10}(time)$) and the problem dimension on the x-axis. Submit files **mem_swaps.hpp** and the plot.

```
void swapRows(std::vector<double> &matrix, int nRows, int nCols, int i, int j);
void swapCols(std::vector<double> &matrix, int nRows, int nCols, int i, int j);
```

   You may test the swap capabilities on randomly selected index pairs using a function like this:

```
#include <utility>   // For std::pair
std::pair<int, int> getRandomIndices(int n)
{
    int i = std::rand() % n;
    int j = std::rand() % (n - 1);
    if (j >= i)
    {
        j++;
    }
    return std::make_pair(i, j);
}
// ... from inside main()
//std::pair<int, int> rowIndices = getRandomIndices(M);
//int i = rowIndices.first;
//int j = rowIndices.second;
//std::pair<int, int> colIndices = getRandomIndices(N);
// ...
```

4. (+10) **Linear algebra**.
   (a) (+5) **Basis**. Does the given set $S = \{1, 1-x, (1-x)^2\}$ form a basis for the space of polynomials up to degree 2, $x \in \mathbb{R}$? Show your work. (Hint: must be both linearly independent and span the space)

(b) (+5) **Gram-Schmidt**. Given set $S = \{x_1(t), x_2(t), x_3(t)\} \ni x_1(t) = t^2, \quad x_2(t) = t, \quad x_3(t) = 1$ and $t \in [-1, 1] \in \mathbb{R}$, construct by hand an orthonormal basis for S using the inner product $(x, y) = \int_{-1}^{1} x(t)y(t)\,dt$.

5. (+20) **Strassen**. Use notes from the class lecture to implement the C++ template for the (recursive) Strassen matrix multiplication algorithm for matrices with even dimension, `C = strassenMultiply(A, B);`. Use the provided starter code `strassen.cpp` and implement in it the declared method `strassenMultiply`. Plot the average performance (FLOPs) multiplying `double` precision square matrices of even dimension from $n = 2$ to $n = 512$ as measured *ntrial* times, $ntrial \geq 3$. Submit `strassen.cpp` and performance plot.