CSC373 Worksheet 1 Solution

July 16, 2020

1.
```
Strassen_Algorithm(A,B):
    n = A.rows
    let C be a new n x n matrix

    if n == 1
        C_11 = A_11 * B_11

    else partition as in step 3 of strassen's algorithm

        p1 = Strassen_Algorithm(A_11, B_12) -
             Strassen_Algorithm(A_11, B_22)

        p2 = Strassen_Algorithm(A_11, B_22) +
             Strassen_Algorithm(A_12, B_22)

        p3 = Strassen_Algorithm(A_21, B_11) +
             Strassen_Algorithm(A_22, B_11)

        p4 = Strassen_Algorithm(A_22, B_21) -
             Strassen_Algorithm(A_22, B_11)

        p5 = Strassen_Algorithm(A_11, B_11) +
             Strassen_Algorithm(A_11, B_22) +
             Strassen_Algorithm(A_22, B_11) +
             Strassen_Algorithm(A_22, B_22)

        p6 = Strassen_Algorithm(A_12, B_21) +
             Strassen_Algorithm(A_12, B_22) -
             Strassen_Algorithm(A_22, B_21) -
             Strassen_Algorithm(A_22, B_22)

        p7 = Strassen_Algorithm(A_11, B_11) +
             Strassen_Algorithm(A_11, B_12) -
             Strassen_Algorithm(A_21, B_11) -
             Strassen_Algorithm(A_21, B_12)

        C_11 = p5 + p4 - p2 + p6
        C_12 = p1 + p2
        C_21 = p3 + p4
```

```
            C_22 = p5 + p1 - p3 - p7

        return C

```

Notes:

- Strassen's method for matrix multiplication
  - Reduces the time complexity of matrix multiplication from $O(n^3)$ to $O(n^{\log_2 7}) = O(n^{2.81})$
  - Has four steps

    1) Divide the input matrics A and B and output matrix C into $n/2 \times n/2$ submatrices

    $$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix},$$

    2) Create 10 matrices, $S_1, S_2, ..., S_{10}$ each of which is $n/2 \times n/2$ and is the sum or difference of two matrices created in step 1

    $S_1 = B_{12} - B_{22}$
    $S_2 = A_{11} + A_{12}$
    $S_3 = A_{21} + A_{22}$
    $S_4 = B_{21} - B_{11}$
    $S_5 = A_{11} + A_{22}$
    $S_6 = B_{11} + B_{22}$
    $S_7 = A_{12} - A_{22}$
    $S_8 = B_{21} + B_{22}$
    $S_9 = A_{11} - A_{21}$
    $S_{10} = B_{11} + B_{12}$

    3) Recursively multiply $n/2 \times n/2$ matrices seven times to compute the following $n/2 \times n/2$ matrices

    $P_1 = A_{11} \cdot S_1 = A_{11} \cdot B_{12} - A_{11} \cdot B_{22}$
    $P_2 = S_2 \cdot B_{22} = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}$
    $P_3 = S_3 \cdot B_{11} = A_{21} \cdot B_{11} + A_{22} \cdot B_{11}$
    $P_4 = A_{22} \cdot S_4 = A_{21} \cdot B_{11} + A_{22} \cdot B_{11}$
    $P_5 = S_5 \cdot S_6 = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}$
    $P_6 = S_7 \cdot S_8 = A_{12} \cdot B_{21} + A12 \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}$
    $P_7 = S_9 \cdot S_{10} = A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12}$

    4) Construct the four $n/2 \times n/2$ submatrices of the product C

    $C_{11} = P_5 + P_4 - P_2 + P_6 = A_{11} \cdot B_{11} + A_{12} + B_{12}$
    $C_{12} = P_1 + P_2 = A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$
    $C_{21} = P_3 + P_4 = A_21 \cdot B_11 + A_{22} \cdot B_{21}$
    $C_{22} = P_5 + P_1 - P_3 - P_7 = A_{22} \cdot B_{22} + A_{21} \cdot B_{12}$

Example: Use Strassen's algorithm to compute the matrix product

$$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

* **STEP 1**

  $A_{11} = 1, A_{12} = 3, A_{21} = 7, A_{22} = 5$

  $B_{11} = 6, B_{12} = 8, B_{21} = 4, B_{22} = 2$

* **STEP 2**

  $S_1 = B_{12} - B_{22} = 4 - 2 = 2$

  $S_2 = A_{11} + A_{12} = 1 + 3 = 4$

  $S_3 = A_{21} + A_{22} = 7 + 5 = 12$

  $S_4 = B_{21} - B_{11} = 4 - 6 = -2$

  $S_5 = A_{11} + A_{22} = 1 + 5 = 6$

  $S_6 = B_{11} + B_{22} = 6 + 2 = 8$

  $S_7 = A_{12} - A_{22} = 3 - 5 = -2$

  $S_8 = B_{21} + B_{22} = 8 + 2 = 10$

  $S_9 = A_{11} - A_{21} = 3 - 5 = -2$

  $S_{10} = B_{11} + B_{12} = 6 + 4 = 10$

* **STEP 3**

  $P_1 = A_{11} \cdot S_1 = A_{11} \cdot B_{12} - A_{11} \cdot B_{22} = 1 \cdot 4 - 1 \cdot 2 = 2$

  $P_2 = S_2 \cdot B_{22} = A_{11} \cdot B_{22} + A_{12} \cdot B_{22} = 1 \cdot 2 + 3 \cdot 2 = 8$

  $P_3 = S_3 \cdot B_{11} = A_{21} \cdot B_{11} + A_{22} \cdot B_{11} = 6 \cdot 7 + 6 \cdot 5 = 72$

  $P_4 = A_{22} \cdot S_4 = A_{22} \cdot B_{21} - A_{22} \cdot B_{11} = 5 \cdot 4 - 5 \cdot 6 = -10$

  $P_5 = S_5 \cdot S_6 = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} = 48$

  $P_6 = S_7 \cdot S_8 = A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} = -20$

  $P_7 = S_9 \cdot S_{10} = A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} = -20$

- **STEP 4**

  $C_{11} = P_5 + P_4 - P_2 + P_6 = 48 - 10 - 8 - 20 = 10$

  $C_{12} = P_1 + P_2 = 10$

  $C_{21} = P_3 + P_4 = 62$

  $C_{22} = P_5 + P_1 - P_3 - P_7 = 48 + 2 - 72 + 20 = -2$

- Is not preferred in practical purposes

  1) The constants used in Strassen's method are high and for a typical application Naive method works better.
  2) For Sparse matrices, there are better methods especially designed for them.
  3) The submatrices in recursion take extra space.
  4) Because of the limited precision of computer arithmetic on noninteger values, larger errors accumulate in Strassen's algorithm than in Naive Method

References:

1) GeeksForGeeks, Divide and Conquer — Set 5 (Strassen's Matrix Multiplication), link

- Regular matrix multiplication

  -

- The master method for solving recurrences
  - provides 'cookbook' method for solving recurrences of the form

$$T(n) = aT(n/b) + f(n)$$

  - depends on the following theorem
    - Let $a \leq 1$ and $b > 1$ be constants, let $f(n)$ be a function and let $T(n)$ be defined on the nonnegative integers by the recurrence

      $T(n) = aT(n/b) + f(n)$

      where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

      1. If $f(n) = O(n^{\log_b a - \epsilon}) for some constant \epsilon > 0, then T(n) = \Theta(n^{\log_b a})$
      2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$
      3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$, and all sufficiently large n, then $T(n) = \Theta(f(n))$.

      Example:

      $T(n) = 9T(n/3) + n$