Design and Analysis of Algorithms Lab
Academic Year: 2020 - 21

Dr. Praveen Kumar Alapati
Sri. G. Brahmaiah (Ph.D. Scholar)
Brahmaiah20pcse001@mahindrauniversity.edu.in
praveenkumar.alapati@mahindrauniversity.edu.in

Department of Computer Science and Engineering

Ecole Centrale School of Engineering

Mahindra University
Global Thinkers. Engaged Leaders.

DAA Lab 5 Due Date: March 14, 2021

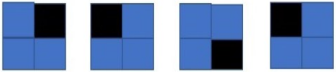
1. Develop a program for the Defective Chessboard problem (N=1024, 2048, and 4096). Use *gettimeofday*() for calculating *runtime* (the average of 5 runs).
2. Develop a program to multiply two square-matrices of order 1024 X 1024 using Strassen's Matrix Multiplication. Use *gettimeofday*() for calculating *runtime* (the average of 5 runs).

Bonus Problem Statements:

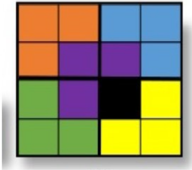
1. Given an array of n numbers and a positive integer i, write a program to find the i^{th} smallest element that runs in O(n) time.
2. Given two sorted arrays, each consisting of n numbers, write a program to find the median of $2n$ elements that runs in $\mathcal{O}(\log n)$ time.

Logic: Defective Chessboard

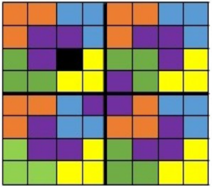
A chessboard that has one unavailable square. We have to cover the remaining squares using triominos.
(Triomino is an **L** shaped object and it is formed with three squares.)



2 x2 2 x2 2 x2 2 x2



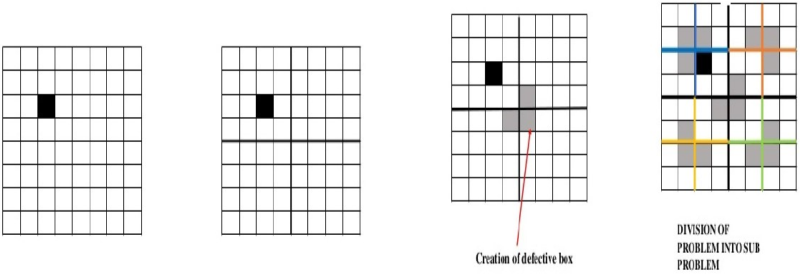
4x4



8x8

Black color square is the defective one.
Number of triomino's required for an $n \times n$ defective chess board: $\frac{n^2-1}{3}$.

8 X 8 Defective Chessboard



1. Divide the chessboard into 4 equal parts.
2. Identify the part which has the defective square and put a triomino that cover all the remaining three parts.
3. Now assume that all 4 parts are defective chessboards.
4. Repeat the steps 1 to 3 until all the squares are covered with triominos.

Defective Chessboard: Analysis

$$
\begin{aligned}
T(n) &= 4 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(1) \\
&= 4 \cdot T\left(\frac{n}{2}\right) + constant \\
&= 4 \cdot T\left(\frac{n}{2}\right) + constant \\
&= \Theta(n^2)
\end{aligned}
$$

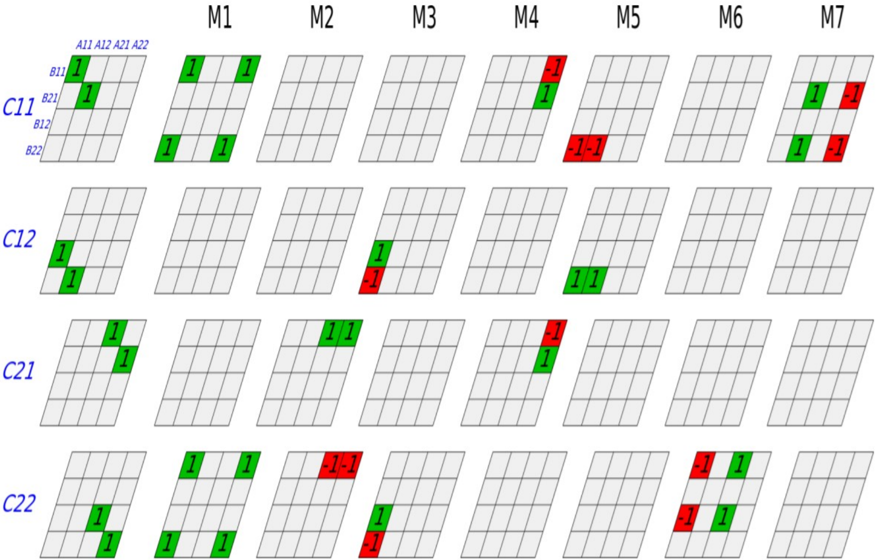
Reasoning:

From case 1 of Master Theorem, where a=4, b=2, and f(n)= $\mathcal{O}(1)$
$n^{\log_b a} = n^{\log_2 4}$
$f(n) = n^{\log_2 4 - \epsilon}$, where $\epsilon = 2$
So, $f(n)$ is polynomially less than $n^{\log_2 4} = n^2$.
$\therefore \; T(n) = \Theta(n^2)$

Logic: Strassen's Matrix Multiplication

$$
\begin{aligned}
M_1 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\
M_2 &= (A_{21} + A_{22}) \cdot B_{11} \\
M_3 &= A_{11} \cdot (B_{12} - B_{22}) \\
M_4 &= A_{22} \cdot (B_{21} - B_{11}) \\
M_5 &= (A_{11} + A_{12}) \cdot B_{22} \\
M_6 &= (A_{21} - A_{11}) \cdot (B_{11} + B_{12}) \\
M_7 &= (A_{12} - A_{22}) \cdot (B_{21} + B_{22})
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= M_1 + M_4 - M_5 + M_7 \\
C_{12} &= M_3 + M_5 \\
C_{21} &= M_2 + M_4 \\
C_{22} &= M_1 - M_2 + M_3 + M_6
\end{aligned}
$$


M1
M2
M3
M4
M5
M6
M7
A11 A12 A21 A22
B11
B21
B12
B22
C11
C12
C21
C22
1
-1

Strassen's Matrix Multiplication:Analysis

$$
\begin{aligned}
T(n) &= 7 \cdot T\left(\frac{n}{2}\right) + 18 \cdot \mathcal{O}\left(\frac{n^2}{4}\right) \\
&= 7 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}\left(n^2\right) \\
&= 7 \cdot T\left(\frac{n}{2}\right) + c \cdot n^2 \\
&= \Theta(n^{2.81})
\end{aligned}
$$

Reasoning:

From case 1 of Master Theorem, where a=7, b=2, and f(n)= $\mathcal{O}(n^2)$
$n^{\log_b a} = n^{\log_2 7}$
$f(n) = n^{\log_2 7-\epsilon}$, where $\epsilon = 0.81$
So, $f(n)$ is polynomially less than $n^{\log_2 7} = n^{2.81}$.
$\therefore\ T(n) = \Theta(n^{2.81})$

DAA Lab Submission Guide Lines

- Mail-ID: cs203.daa.mec@gmail.com (Doubt Clarification).
- Submission Link will be shared.
- Late Submission (<=3-Days):50% weightage will be given.
- Write a readme file to understand your solutions.
- Submit source files only (C or JAVA).

Lab Weightage - 30%.
Lab Instructor: Sri. Brahmaiah G

DAA (Design and Analysis of Algorithms) Lab

Reference Books:

1. Introduction to Algorithms, 3rd edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekaran.
3. Algorithms, 4th edition, Robert Sedgewick.
4. Design and Analysis of Computer Algorithms, Aho, Ullman, and Hopcroft.

Web Resources:

1. Algorithms by Robert Sedgewik
2. Algorithms by Abdul Bari
3. MIT - Open Courseware Videos on Algorithms
4. Data Structures and Algorithms