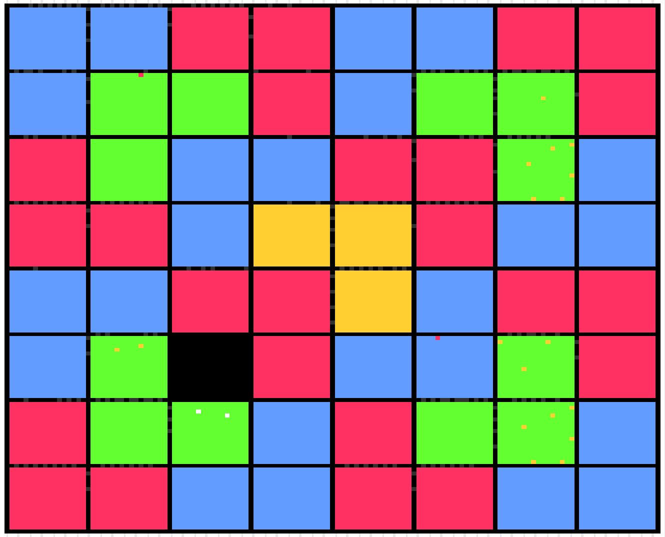
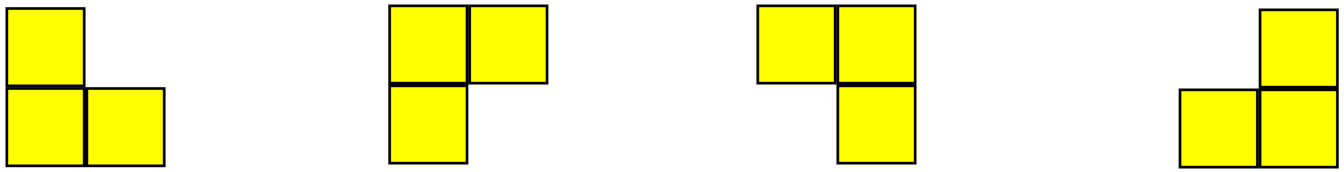
Defective Chessboard

Dr. Bibhudatta Sahoo
Communication & Computing Group
Department of CSE, NIT Rourkela
Email: bdsahu@nitrkl.ac.in, 9937324437, 2462358



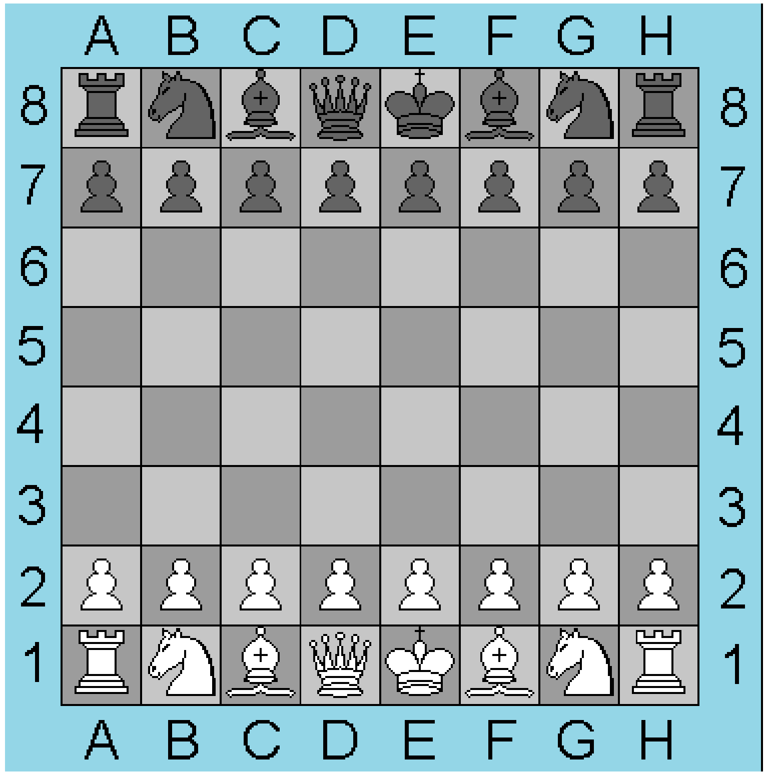
Problem Statement

- The **Defective Chessboard** problem, also known as the Tiling Problem
- **Input**: A **n × n** board where n is of form 2^k where $k \geq 1$ (Basically n is a power of 2 with minimum value as 2). The board has **one missing cell** of size 1 × 1.

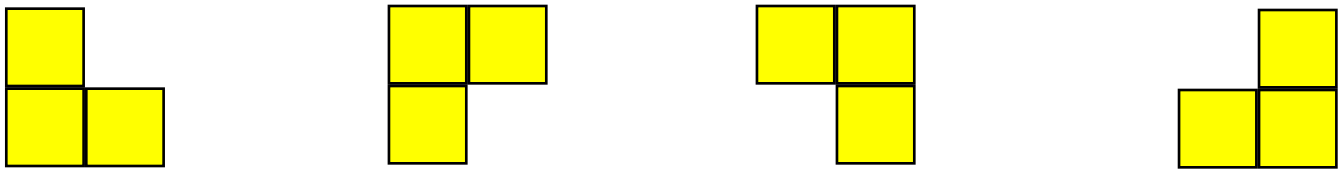


- Fill the board using **L** shaped tiles.
- A L shaped tile is a **2 × 2** square with one cell of size 1×1 missing.
- **Output:** A tiling of the board using a triomino, a three square tile obtained by deleting the upper right 1 by 1 corner from a 2 by 2 square.
- You are allowed to rotate the triomino, for tiling the board

Tiling A Defective Chessboard



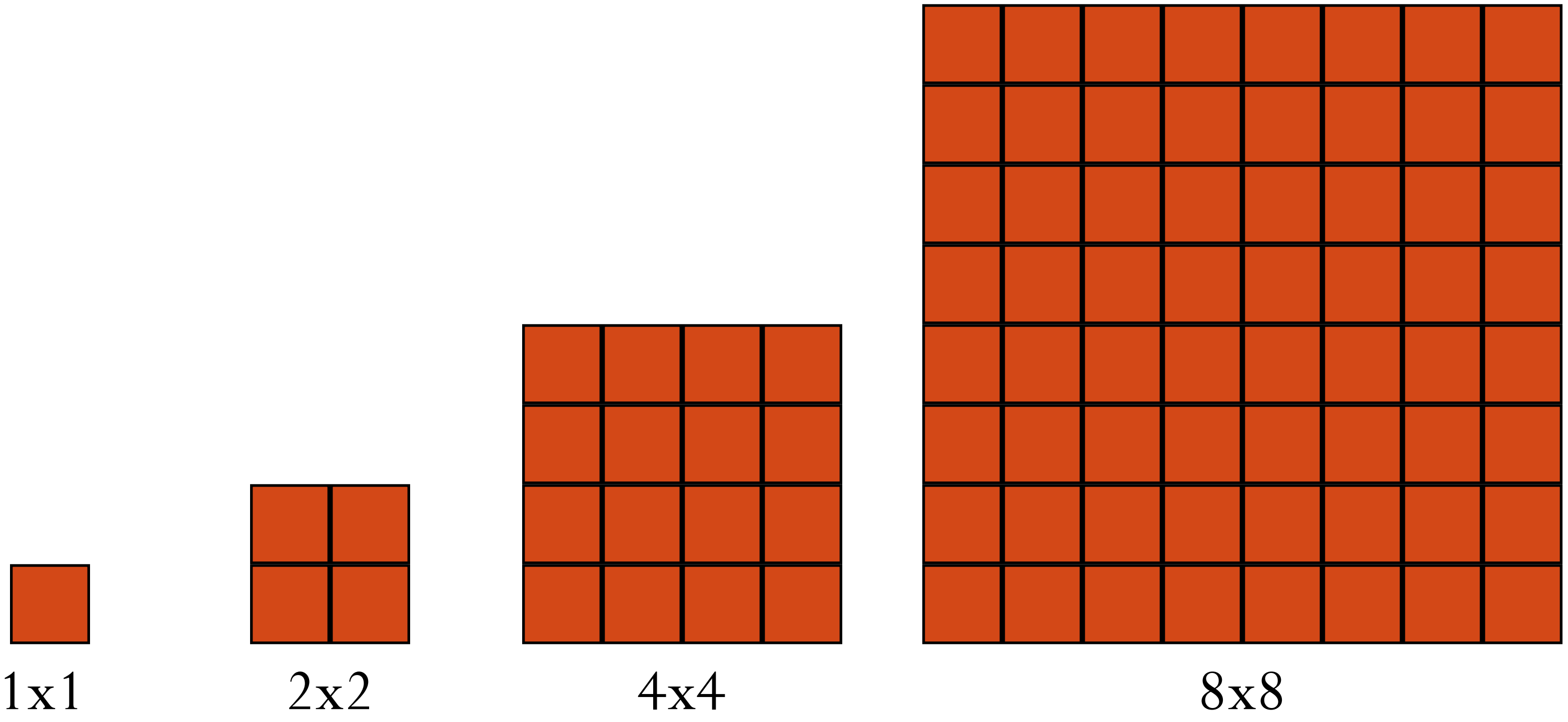
Fill the board using **L** shaped tiles.



Triominoes with different orientation

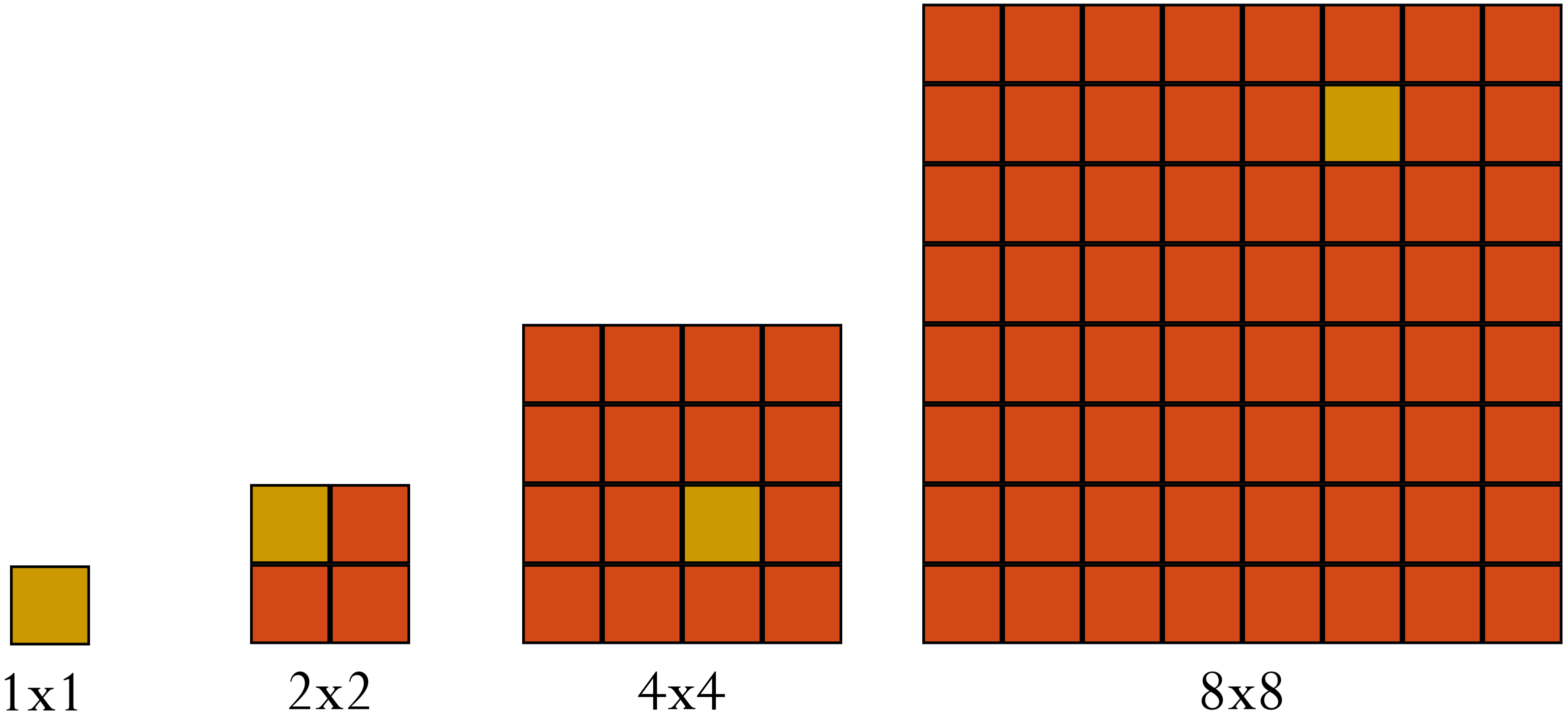
Our Definition of A Chessboard

A chessboard is an **n** × **n** grid, where **n** is a power of **2**.



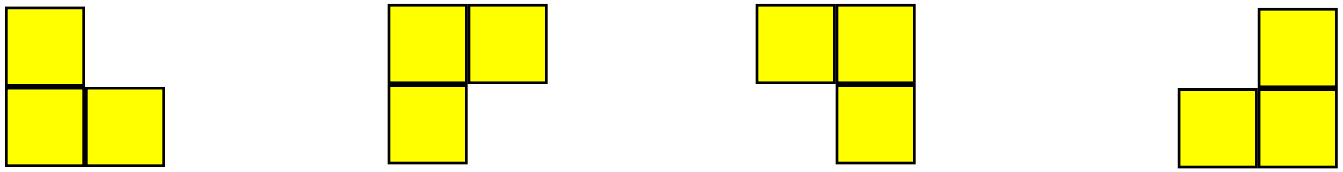
A Defective Chessboard

- A defective chessboard is a chessboard that has one unavailable (defective) position.



A Triomino

- A triomino is an L shaped object that can cover three squares of a chessboard.
- A triomino has four orientations.

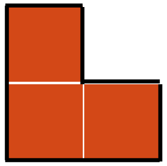


Triominoes with different orientation

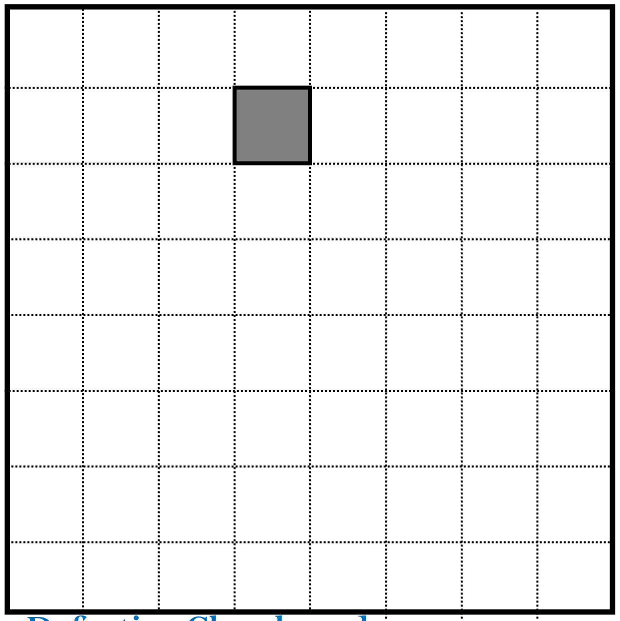
Place $(n^2 - 1)/3$ triominoes on an **n×n defective chessboard** so that all $(n^2 - 1)$ non-defective positions are covered. Recursively **tile** the four **defective 4 × 4 chessboards**.

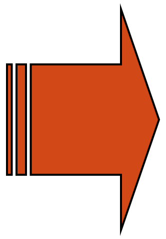
Tromino Tiling

A triomino tile:

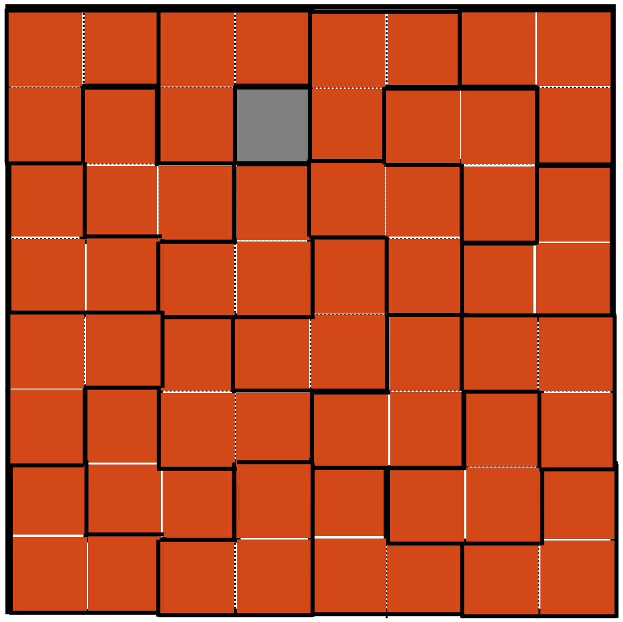


And a 2^nx2^n board with a hole:



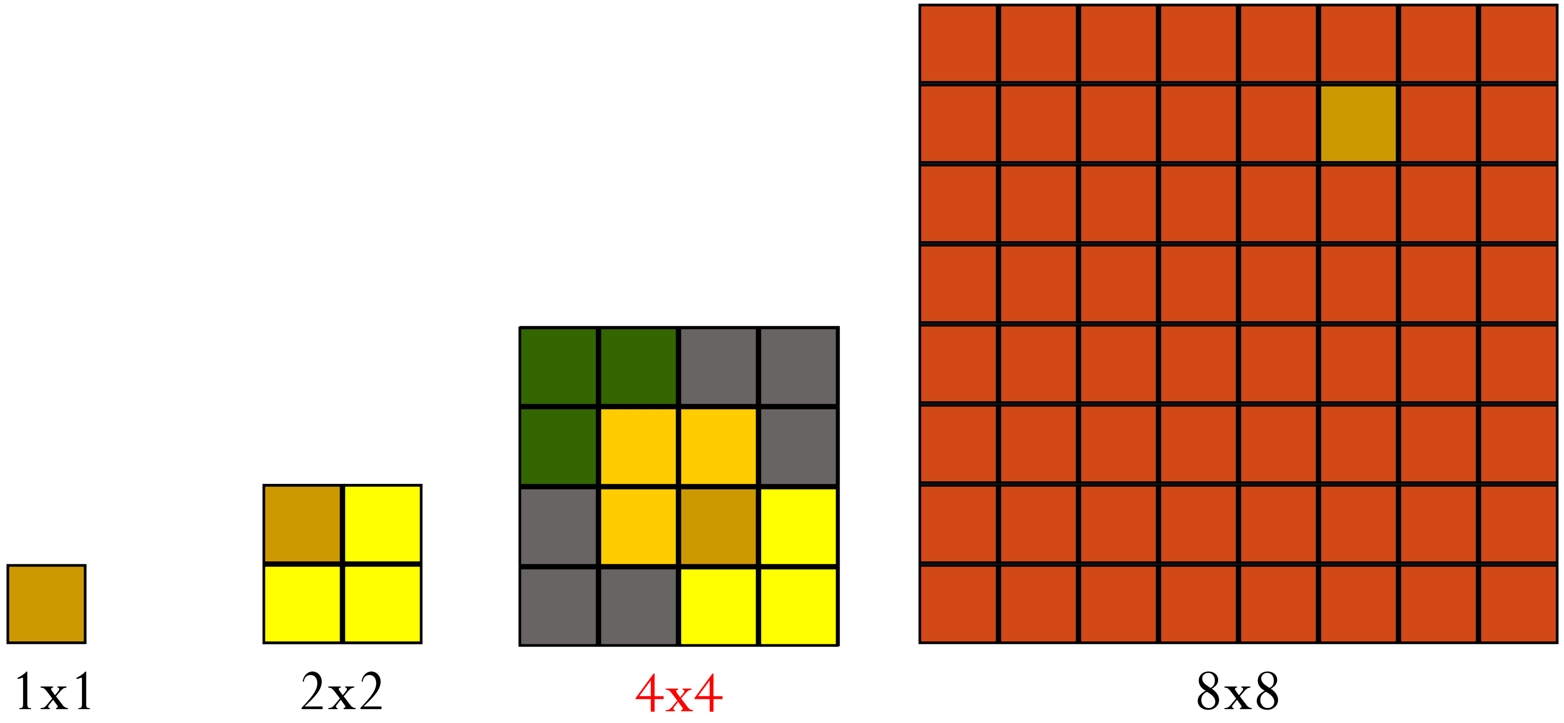


A tiling of the board with triominos:



Tiling A Defective Chessboard

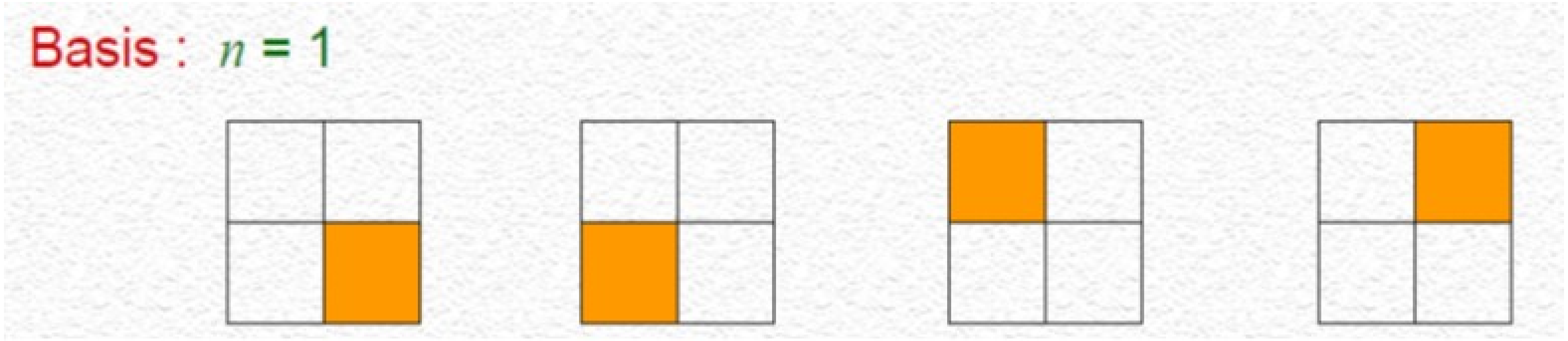
- Place $(n^2 - 1)/3$ triominoes on an $n \times n$ defective chessboard so that all $n^2 - 1$ non-defective positions are covered.



Proof by Induction

- Any 8×8 defective chessboard can be covered with twenty-one triominoes
- **Any $2^n \times 2^n$ defective chessboard can be covered with $\underline{(2^{2n} - 1) / 3}$ triominoes.**

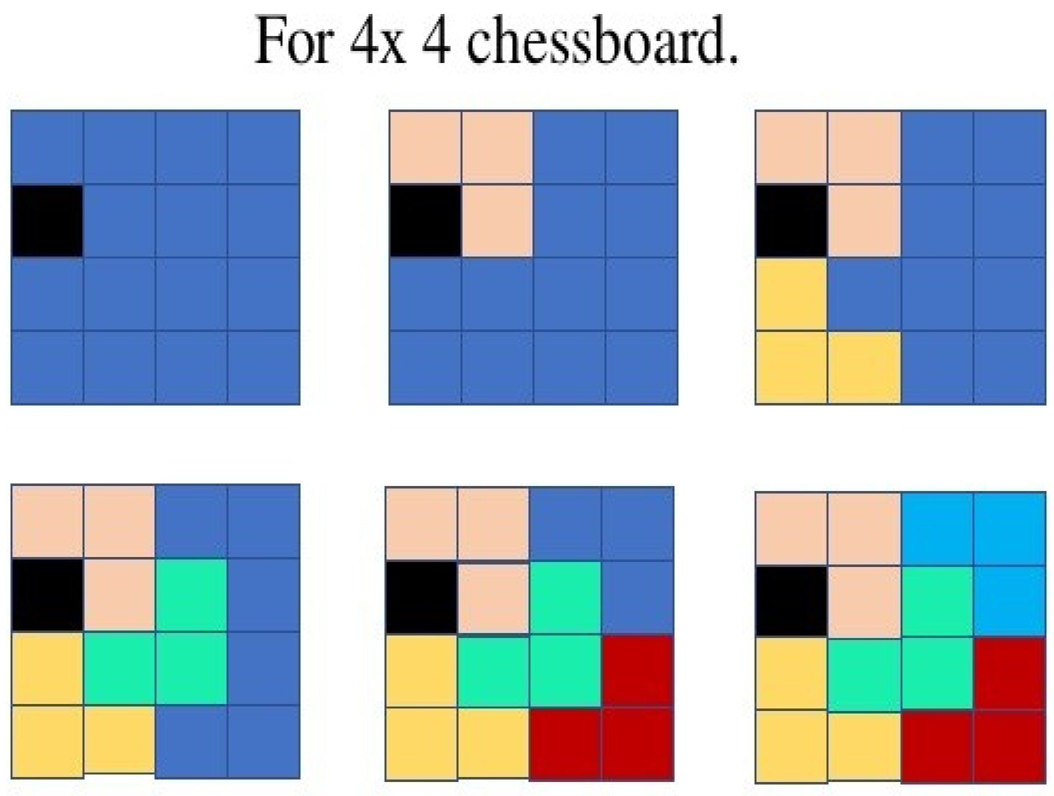
Proof:



- All four possibilities ; one triominoe is required.
- Hence **$(2^{2n} - 1) / 3 = 1$ for $n=1$.**

Proof by Induction: $2^n \times 2^n$ defective chessboard can be covered with $(2^{2n} - 1)/3$ triominoes

For n = 2, A 4×4 chess board requires $(2^{2n} - 1)/3 = 5$ number of Triomino to cover the defective chessboard.



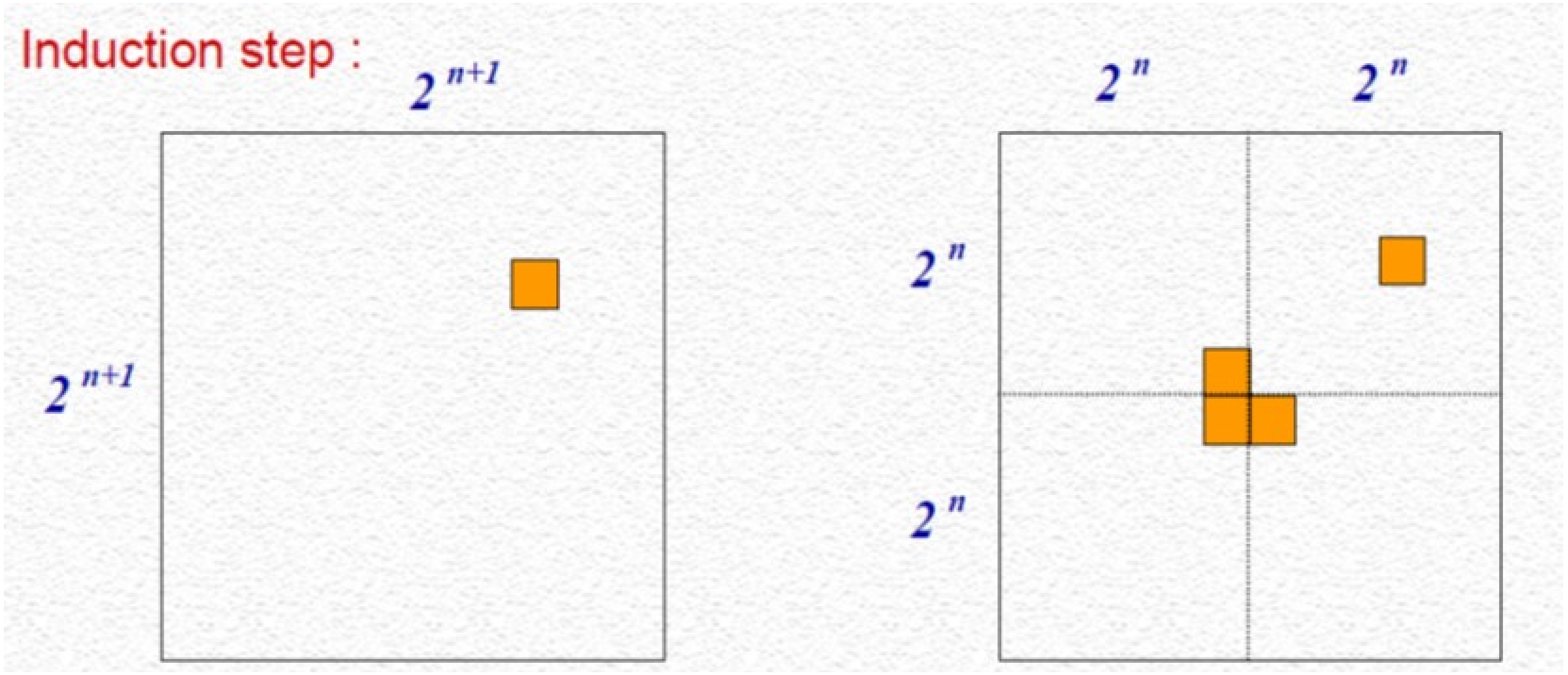
Proof by Induction: $2^n \times 2^n$ defective chessboard can be covered with $(2^{2n} - 1)/3$ triominoes

- Assume that induction hypothesis is true for any **n; n > 0.**

Now to prove hypothesis is true for n+1

$4 \times (2^{2n} - 1)/3 + 1 = 2^2 \times (2^{2n} - 1)/3 + 1 = (2^{2n+1} - 4)/3 + 1$

$= (2^{2n+1} - 4 + 3)/3 = (2^{2n+1} - 1)/3$ □



Tiling A Defective Chessboard

- A simple algorithm about the chessboard coverage problem can be designed by the divide-and-conquer strategy. When $k > 0$, divide the chessboard of $2^k \times 2^k$ into 4 sub-chessboards of $2^{k-1} \times 2^{k-1}$
- The special pane must be located in the small one of four sub-chessboards, and other three sub-chessboards have no special panes.
- To convert these three sub-chessboards into the special chessboard, one L-type triomino can be put on the junction of three smaller chessboards .
- The pane which is covered by the L-type triomino on three sub-chessboards is the special pane on this chessboard, so the original problem is converted into four smaller chessboard coverage problems.
- Recursively use this division until the chessboard is simplified as the chessboard of 1×1.

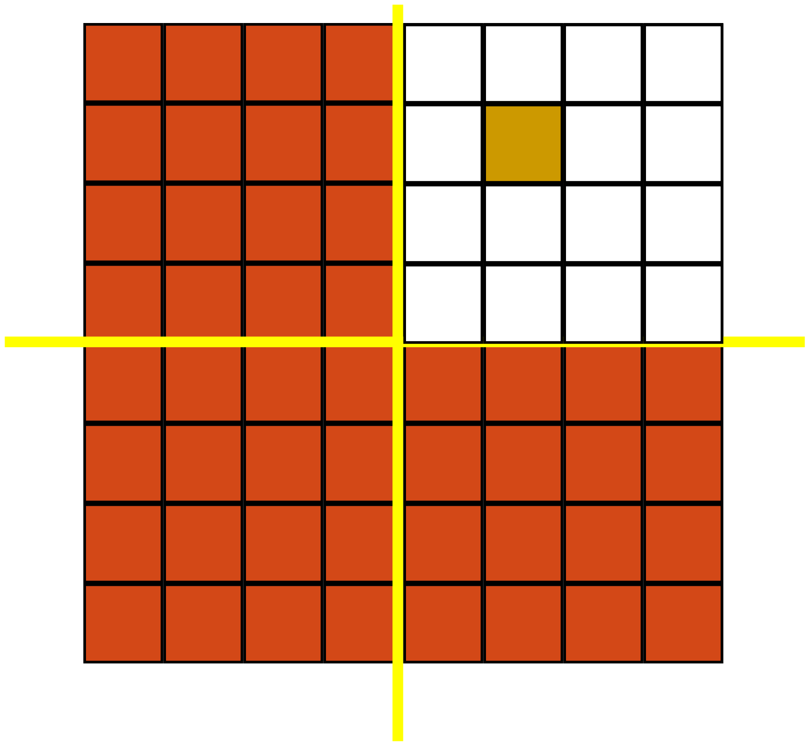
Tiling: Divide-and-Conquer

Tiling is a divide-and-conquer algorithm:

1. Just do it trivially if the board is 2x2, else:
2. **Divide** the board into four smaller boards (introduce holes at the corners of the three smaller boards to make them look like original problems)
3. **Conquer** using the same algorithm recursively
4. **Combine** by placing a single triomino in the center to cover the **three introduced** holes

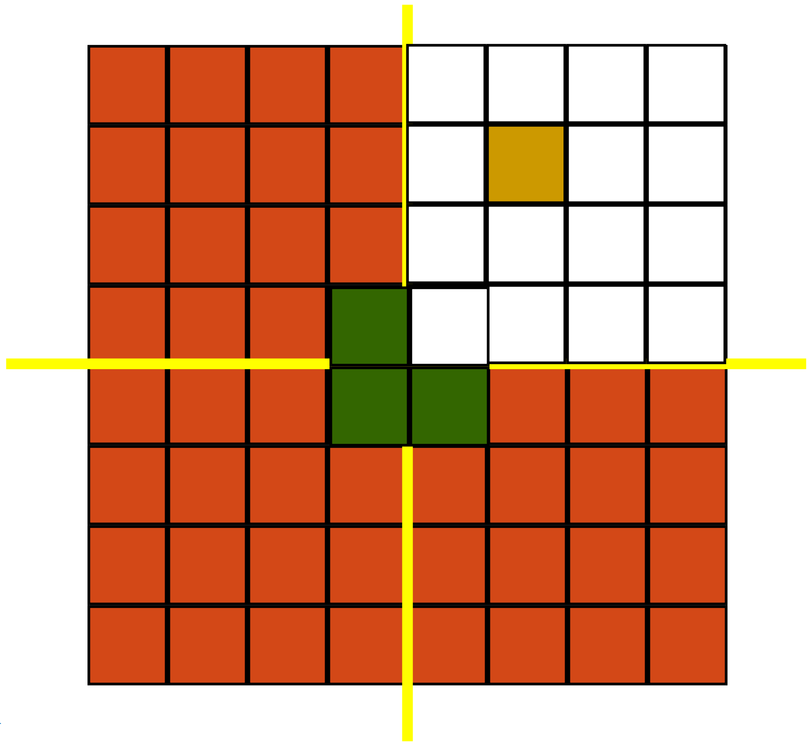
Tiling A Defective Chessboard

- Divide into four smaller chessboards. 4 x 4
- One of these is a defective 4 x 4 chessboard.



Tiling A Defective Chessboard

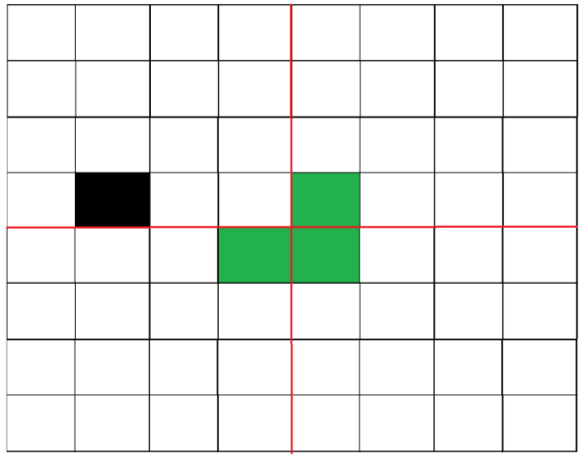
- Make the other three 4 x 4 chessboards defective by placing a triomino at their common corner.
- Recursively tile the four defective 4 x 4 chessboards.



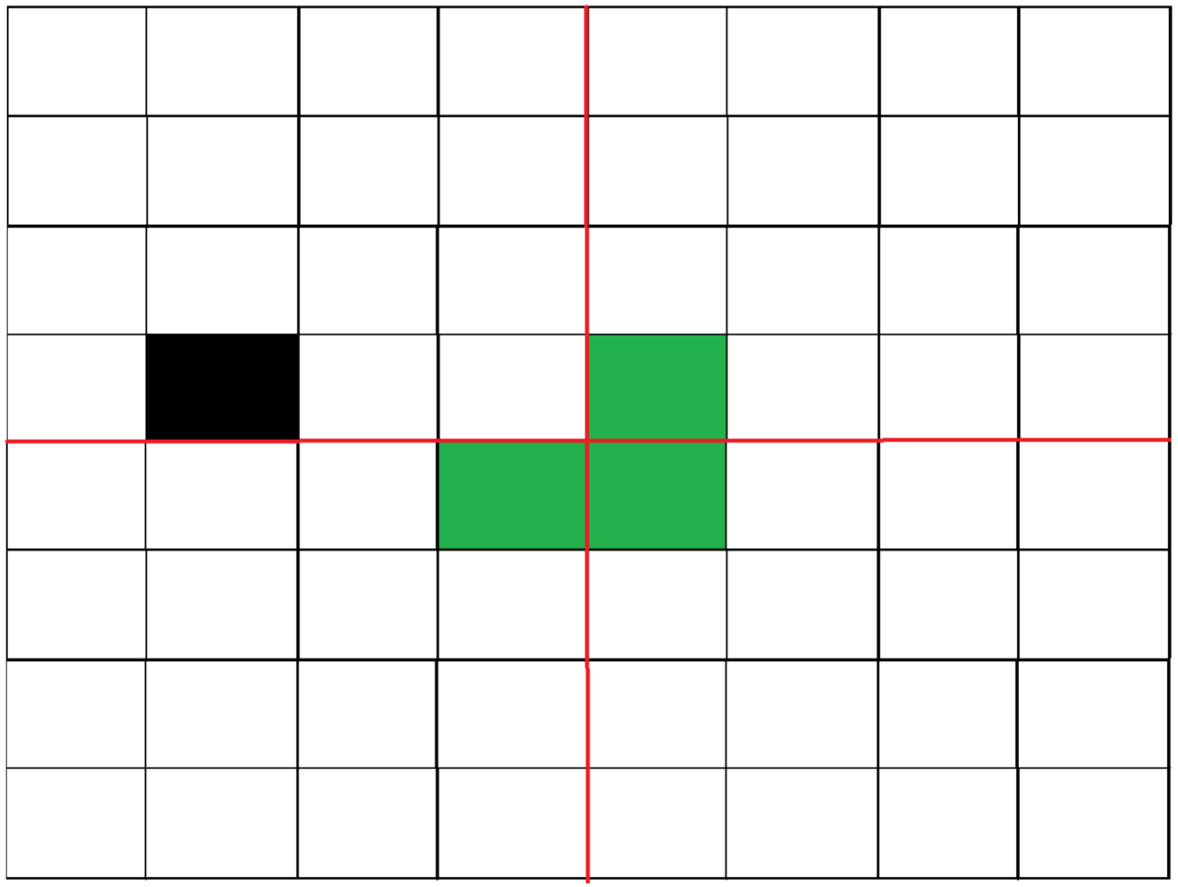
Divide and Conquer : recursive algorithm.

- // n is size of given square, p is location of missing cell

1. Tile(int n, Point p)
2. Base case: n = 2, A 2 × 2 square with one cell missing is nothing but a tile and can be filled with a single tile(Triomino).
3. Place a L shaped tile(Triomino) at the center such that it does not cover the n/2 × n/2 subsquare that has a missing square. **Now all four subsquares of size n/2 × n/2 have a missing cell** (a cell that doesn't need to be filled).
4. Solve the problem recursively for following four. Let p1, p2, p3 and p4 be positions of the 4 missing cells in 4 squares. a) Tile(n/2, p1) b) Tile(n/2, p2) c) Tile(n/2, p3) d) Tile(n/2, p3)

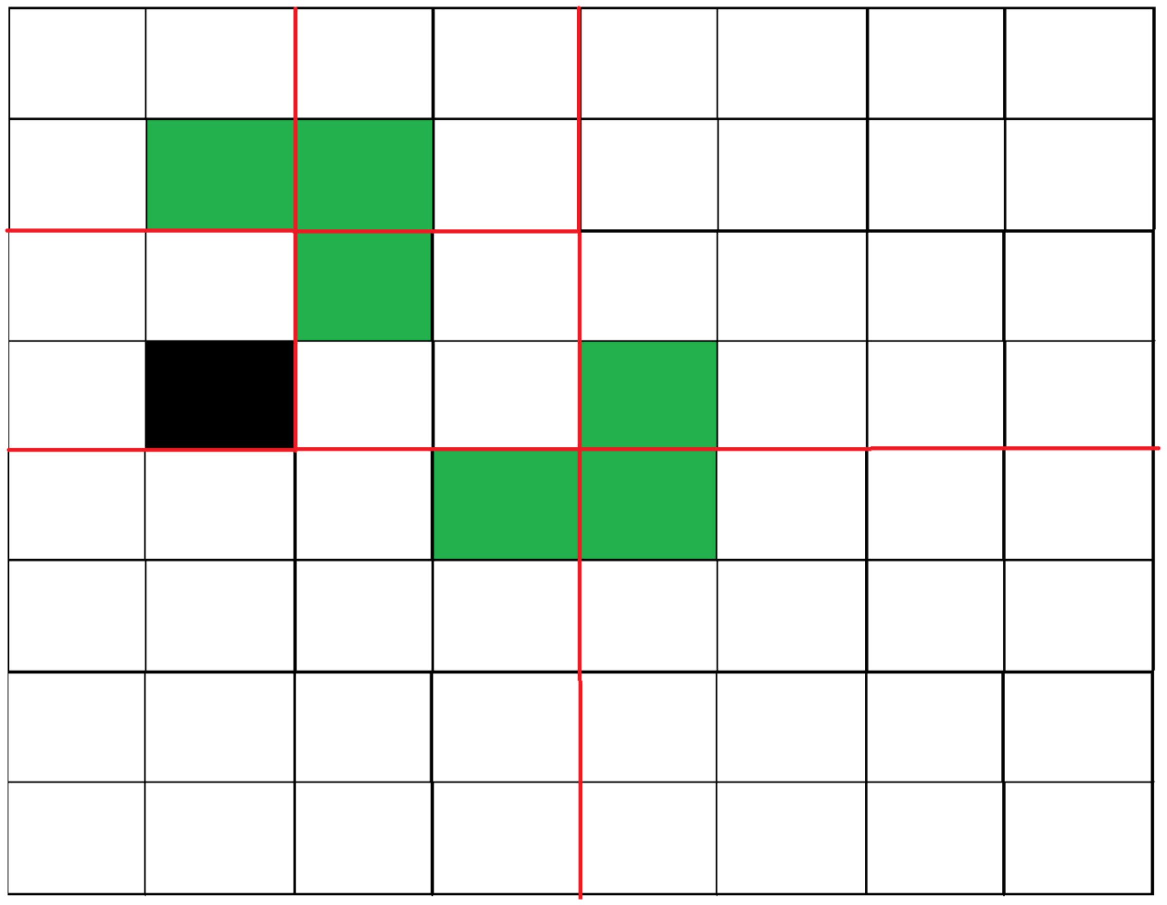


After placing first tile

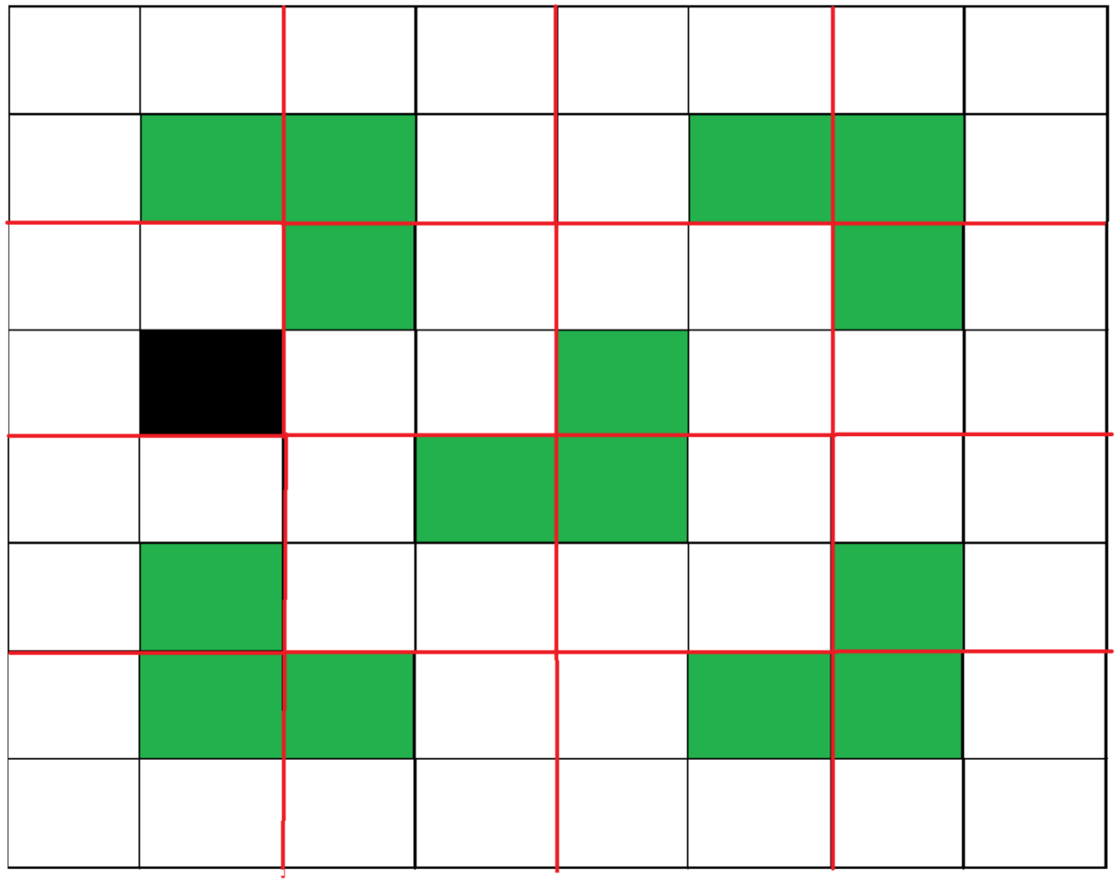


Place a triomino at the center so that it fully covers one square from each of the three (3) subboards with no missing square, and misses the fourth subboard completely.

Recurring for first subsquare.



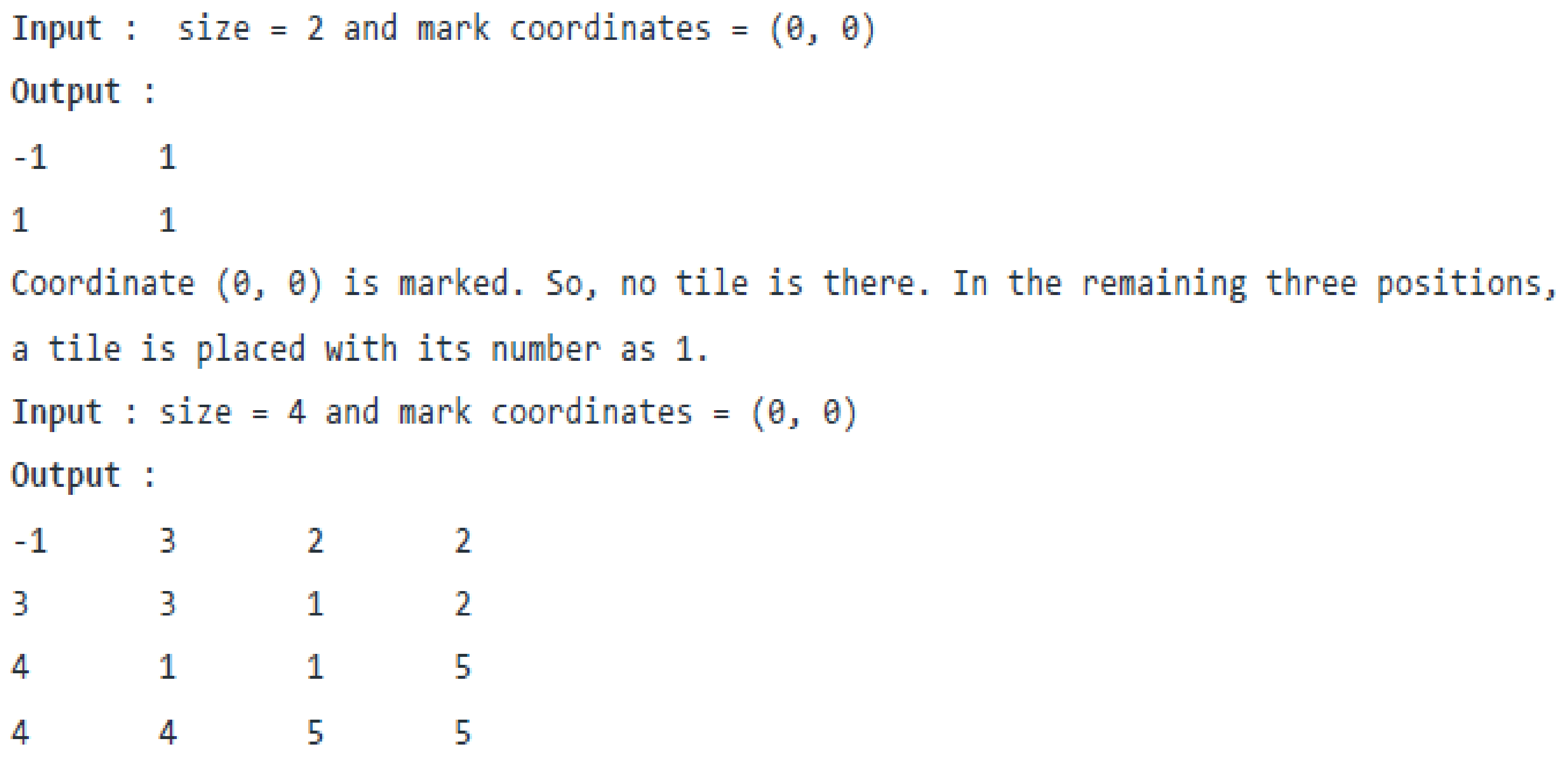
Shows first step in all four subsquares.



Solve each smaller subproblem recursively using the same technique.

Exercise: Algorithms Horowitz, Sahni & Rajasekaran : page 145

- Write a complete program for the defective-chessboard problem. Include modules to welcome the user to the program; Input chess board size and location of the defect; and output the tiled chessboard.



```
Input :  size = 2 and mark coordinates = (0, 0)
Output :
-1      1
1       1
Coordinate (0, 0) is marked. So, no tile is there. In the remaining three positions,
a tile is placed with its number as 1.
Input : size = 4 and mark coordinates = (0, 0)
Output :
-1      3       2       2
3       3       1       2
4       1       1       5
4       4       5       5
```

Algorithm TileBoard

```
Algorithm TileBoard(toprow, topcol, defectrow, defectcol,size)
// toprow is row number of top-left corner of board
// topcol is column number of top-left cornor of board
// defectrow is row  number of defective square.
// defectcol is column number of defective square.
// size is length of one side of chess board
{
If(size = 1) return;
Tiletouse := tile ++;
Quadrantsize := size/2;
// tile top-left quadrant
If (defectrow < toprow + quadrantsize &&
Defectcol < topcol + quadrantsize) then
// defect is in this quadrant
TileBoard (toprow, topcol, defectrow, defectcol, quadrantsize);
else
```

Algorithm: TileBoard

```
17. {
18. // no defect in this quadrant
19. // place a tile in bottom-right cornor
20. Board[toprow + quadrantsize -1][topcol + quadrantsize -1] :=
    tiletouse;
21. //Tile the rest
22. TileBoard(toprow, topcol, toprow + quadrantsize -1, topcol +
    quadrantsize -1, quadrantsize );
23. }
24. // Code for remaining three quadrant is similar
25. }
```

C program : ChessBoard #1/2

```
void ChessBoard(int tr, int tc, int dr, int dc, int size)
{
if(size==1) return;
int t=tile++, s=size/2;
// cover the sub-chessboard of the top left corner
if(dr<tr+s && dc<tc+s)
ChessBoard(tr, tc, dr, dc, s);
else{Board[tr+s-1][tc+s-1]=t;
ChessBoard(tr, tc, tr+s-1, tc+s-1, s); }
// cover the sub-chessboard of the top right corner
if(dr<tr+s && dc>=tc+s)
ChessBoard(tr, tc+s, dr, dc, s);
else{Board[tr+s-1][tc+s]=t;
ChessBoard(tr, tc+s, tr+s-1, tc+s, s); }
```

C program : ChessBoard #2/2

```
// cover the sub-chessboard of the down left corner
if(dr>=tr+s && dc<tc+s)
ChessBoard(tr+s, tc, dr, dc, s);
else{Board[tr+s][tc+s-1]=t;
ChessBoard(tr+s, tc, tr+s, tc+s-1, s); }
// cover the sub-chessboard of the down right corner
if(dr>=tr+s && dc>=tc+s)
ChessBoard(tr+s, tc+s, dr, dc, s);
else{Board[tr+s][tc+s]=t;
ChessBoard(tr+s, tc+s, tr+s, tc+s, s); }
}
```

- In above algorithm, a two-dimensional integer array Board is used to denote the chessboard. Board[0][0] is the pane of **top left corner** of the chessboard.
- Tile is a comprehensive integer variable in the algorithm and it is used to denote the number of the L-type domino, and its initial value is 0.
- And the input parameters of the algorithm include

tr (the row number of the pane of top left corner of the chessboard),

tc (the column number of the pane of top left corner of the chessboard),

dr (the row number of the special pane),

dc (the column number of the special pane),

- size (size=2^k, and it denotes that the specification of chessboard is $2^k \times 2^k$).

A 4×4 Chessboard

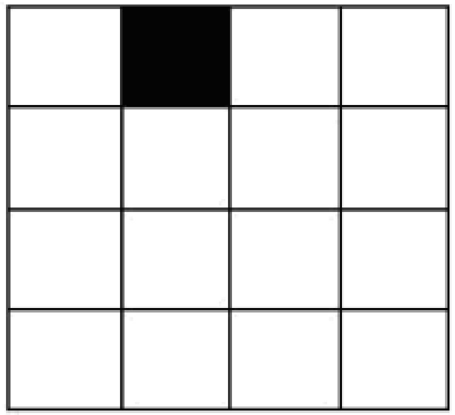


Figure 1. A Special Chessboard when k=2

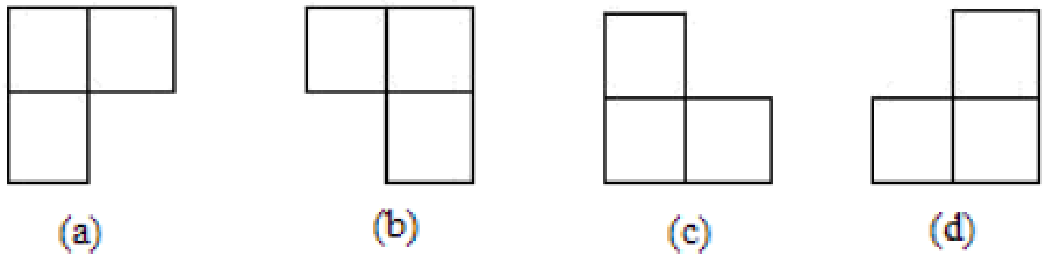
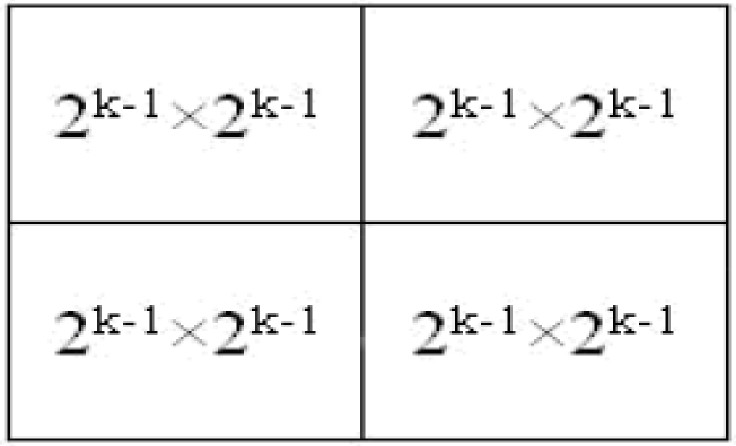
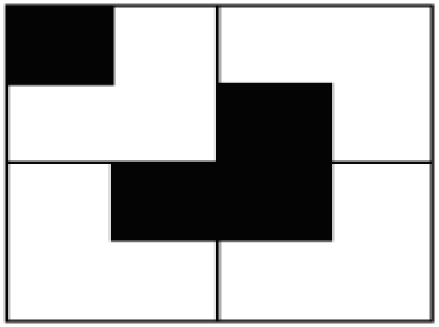


Figure 2. Four L-type Dominos with Different Forms



(a)



(b)

Figure 3. Division of Chessboard

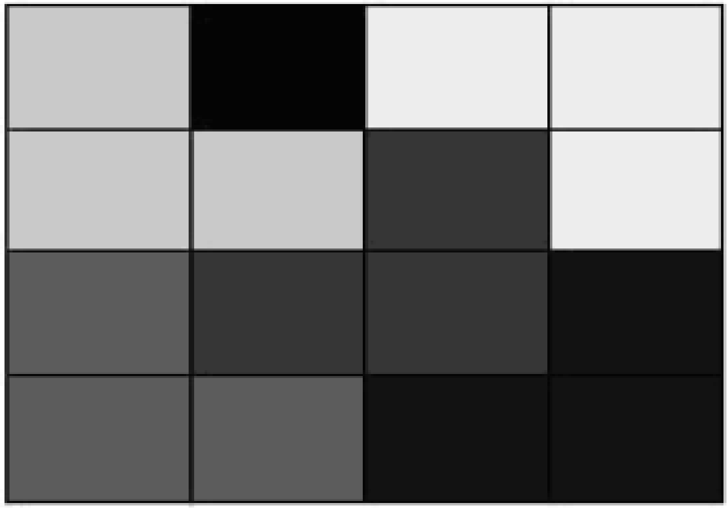


Figure 4. Result of Chessboard Coverage

Chessboard converge results

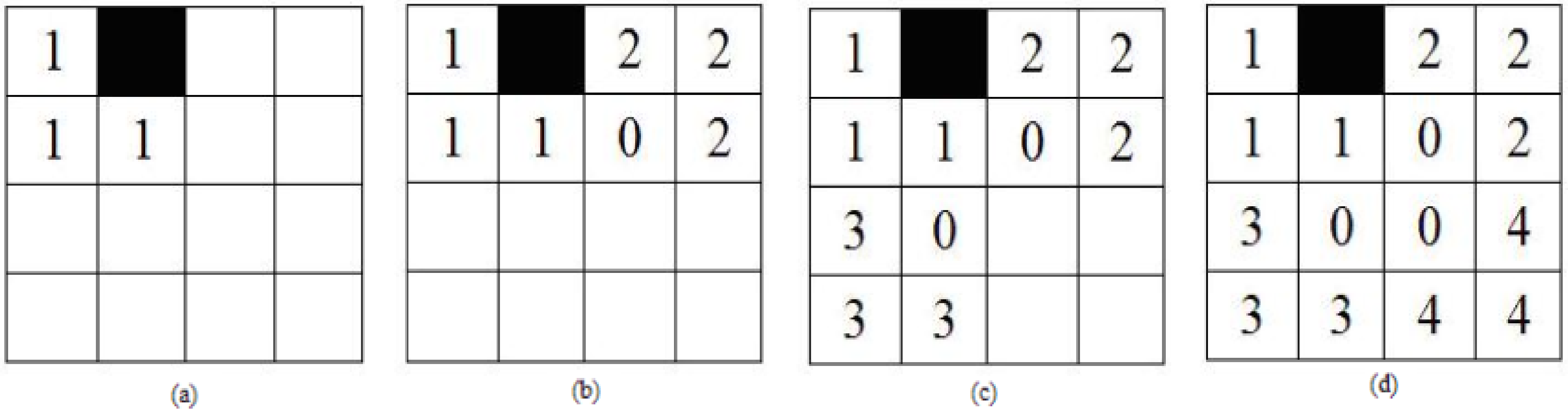


Figure 5. Chessboard Coverage Sequence of the ChessBoard Algorithm

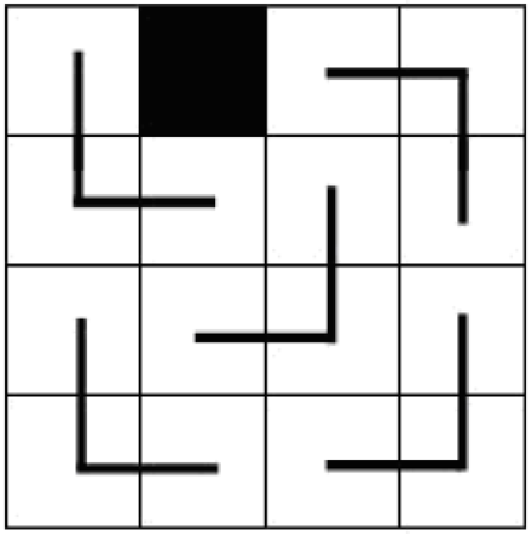


Figure 6. Simple Denotation of Chessboard Coverage Result

Chessboard Coverage Sequence

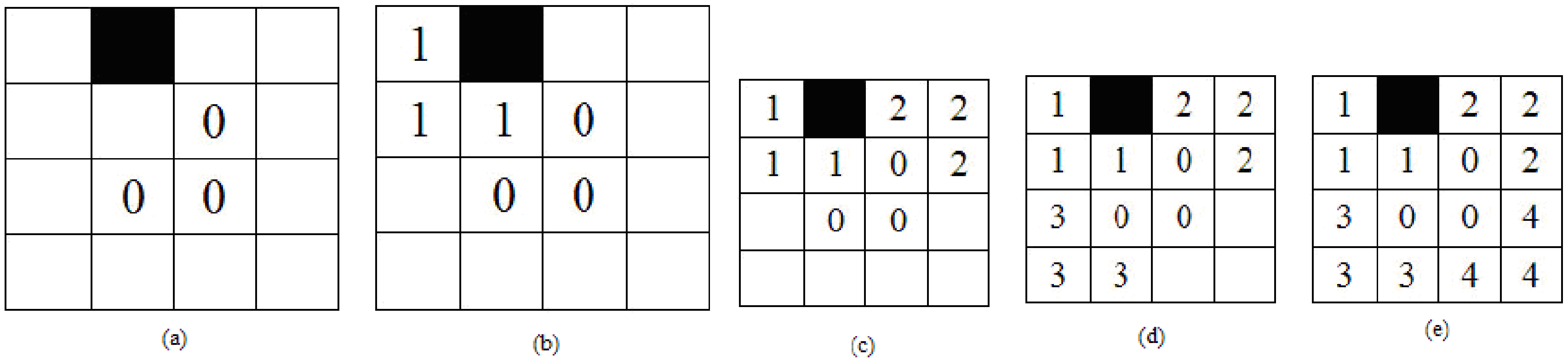


Figure 7. Chessboard Coverage Sequence of the ChessBoard Algorithm

Complexity

- Let $n = 2^k$.
- Let $t(k)$ be the time taken to tile a 2^k x 2^k defective chessboard.
- **$t(0) = d$, where d is a constant.**
- **$t(k) = 4t(k-1) + c$, when $k > 0$. Here c is a constant.**

Recurrence equation for time complexity t().

$t(0) = d$, when $k=0$; here d is a constant.

$t(k) = 4t(k-1) + c$, when $k > 0$; here c is a constant.

Substitution Method

$$
\begin{aligned}
\mathbf{t(k) = 4t(k-1) + c}\\
&= 4[4t(k-2) + c] + c\\
&= 4^2\, t(k-2) + 4c + c\\
&= 4^2[4t(k-3) + c] + 4c + c\\
&= 4^3\, t(k-3) + 4^2c + 4c + c\\
&= \ldots\\
&= 4^k\, t(0) + 4^{k-1}c + 4^{k-2}c + \ldots + 4^2c + 4c + c\\
&= 4^k\, d + 4^{k-1}c + 4^{k-2}c + \ldots + 4^2c + 4c + c\\
&= \Theta(4^k)\\
&= \Theta(\text{number of triominoes placed})
\end{aligned}
$$

- Since we must spend at least $\Theta(1)$ time placing each tile, we cannot obtain an asymptotically faster algorithm than divide and conquer.
- Because the amount of L-type domino to cover one chessboard of $2^k \times 2^k$ is $(4^k-1)/3$, the algorithm of **TileBoard** is the optimal algorithm on the asymptotic meaning.

Conclusion

- Three approaches of the divide-and-conquer algorithm to solve the chessboard coverage problem are an organic integer, and the division of the original problem into many sub-problems is the key content, and it is the key approach to solve problem by the divide-and-conquer algorithm.
- The present divide-and-conquer method uses program skill to solve the problem of chessboard coverage, and splits the L-type dominos and cover, which is not consistent with the former analysis, and adds difficulties for the divide-and-conquer algorithm which is not complex originally, and the teaching effect is not ideal.
- The divide-and-conquer method contains abundant mathematical ideas which should be dug and studied in the teaching to ensure the continuity and consistence of students' thinking development and enhance the learning effect.
- Therefore, aiming at the problem of chessboard coverage, the coverage sequence of L-type dominos in the flow of the divide-and-conquer algorithm is improved, and the improved algorithm can not only eliminate the algorithm skill, but add the standardization and consistence of the algorithm and achieve better teaching effect

Thanks for Your Attention!

Exercises

1. A rectangular floor measuring 'm' feet by 'n' feet is tiled with one-foot square tiles. How many tiles would the diagonal of the rectangle cross?
2. How many square boxes are crossed by a diagonal in a rectangular table formed by 199*991 identical squares?
3. Give the problem definition of a defective chessboard? Explain clearly how divide and conquer method is used to find the solution.
4. Consider an m × n rectangular chessboard. Suppose we want to tile this board with dominoes, where a domino is a 2 × 1 rectangle, and a tiling is a way to place several dominoes on the board so that all of its squares are covered but no dominos overlap or lie partially off the board. Is such a tiling possible? If so, how many are there?

More problems …

1. Measurements: How many orange ping pong balls fit into a square meter box?
2. Why is the shape of books rectangular and not square?
3. Is it possible to install a square or rectangular manhole?
4. In a closed rectangular box with square ends, the total surface area is 216cm^2. What are the dimensions that will give the greatest volume?
5. How many small squares can fit into this big square?
6. Why are manholes circular in shape not rectangular or square?
7. Why are the houses mostly square or rectangular in shape?
8. How many rectangles can be made out of a 3×3 box of squares? Measurements: How many square meters are in a square foot?