GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

Divide and Conquer | Set 6 (Tiling Problem)

Given a n by n board where n is of form 2^k where k >= 1 (Basically n is a power of 2 with minimum value as 2). The board has one missing cell (of size 1 x 1). Fill the board using L shaped tiles. A L shaped tile is a 2 x 2 square with one cell of size 1×1 missing.

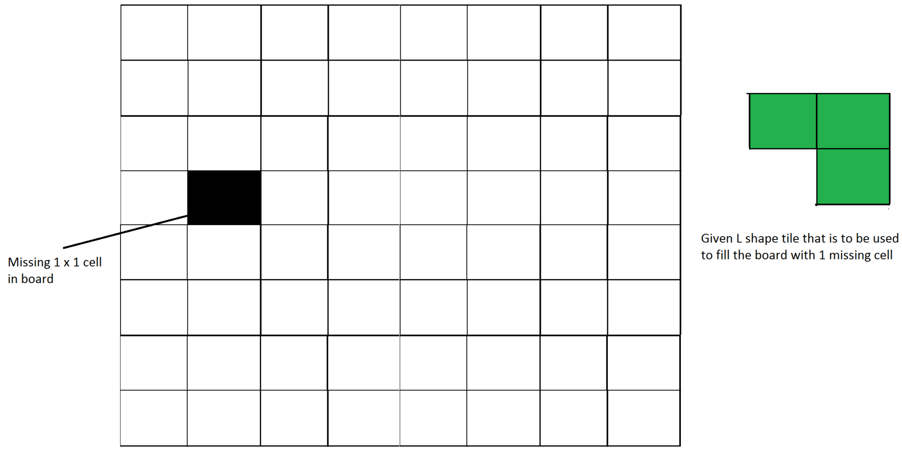

Figure 1: An example input

This problem can be solved using Divide and Conquer. Below is the recursive algorithm.

```
// n is size of given square, p is location of missing cell
Tile(int n, Point p)

1) Base case: n = 2, A 2 x 2 square with one cell missing is nothing
   but a tile and can be filled with a single tile.

2) Place a L shaped tile 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 x n/2 have a missing cell (a cell that doesn't
   need to be filled).  See figure 2 below.

3) 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)
```

The below diagrams show working of above algorithm

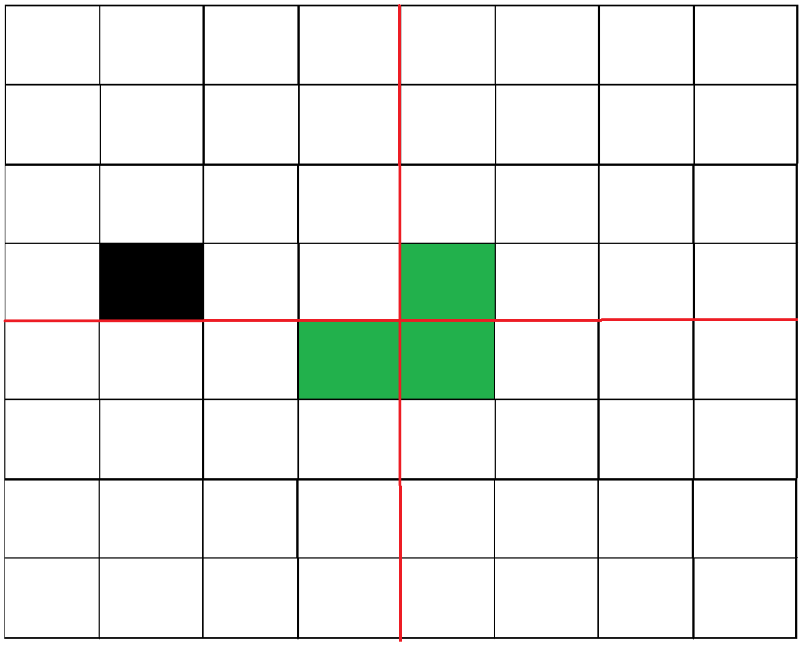

Figure 2: After placing first tile

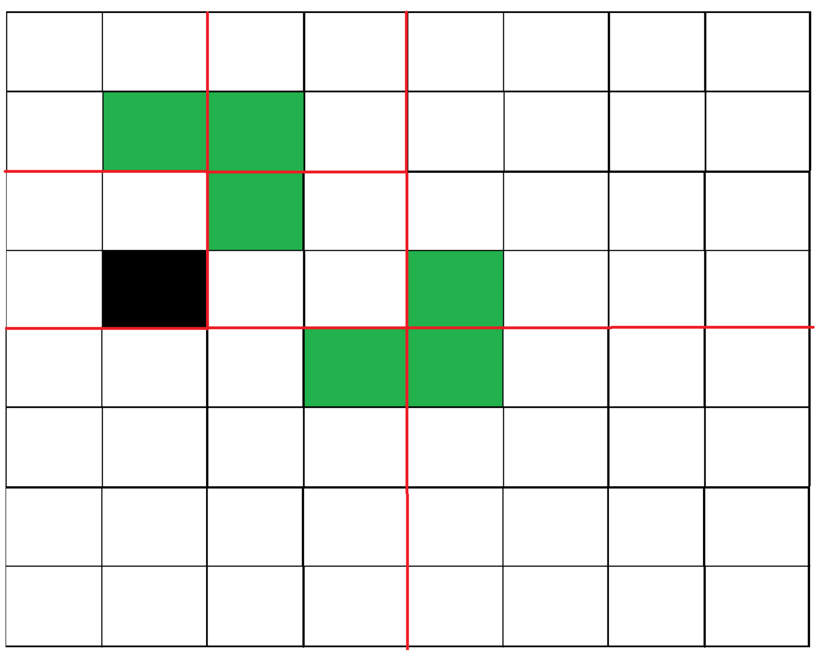

Figure 3: Recurring for first subsquare.

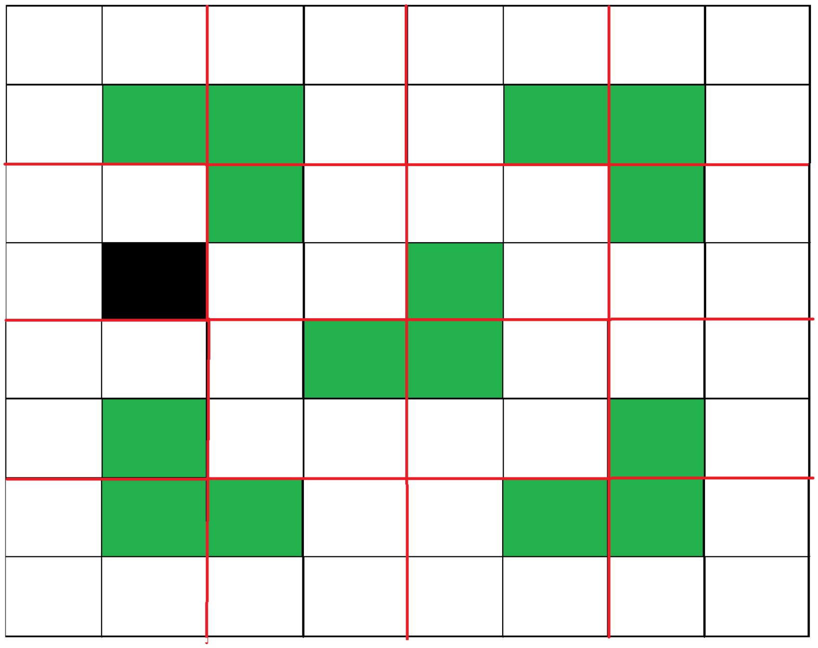

Figure 4: Shows first step in all four subsquares.

Time Complexity:
Recurrence relation for above recursive algorithm can be written as below. C is a constant.
T(n) = 4T(n/2) + C
The above recursion can be solved using Master Method and time complexity is $O(n^2)$

How does this work?
The working of Divide and Conquer algorithm can be proved using Mathematical Induction. Let the input square be of size 2^k x 2^k where k >=1.
Base Case: We know that the problem can be solved for k = 1. We have a 2 x 2 square with one cell missing.
Induction Hypothesis: Let the problem can be solved for k-1.
Now we need to prove to prove that the problem can be solved for k if it can be solved for k-1. For k, we put a L shaped tile in middle and we have four subsqures with dimension 2^{k-1} x 2^{k-1} as shown in figure 2 above. So if we can solve 4 subsquares, we can solve the complete square.

References:
http://www.comp.nus.edu.sg/~sanjay/cs3230/dandc.pdf

This article is contributed by **Abhay Rathi**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Find Union and Intersection of two unsorted arrays
- Pythagorean Triplet in an array
- Maximum profit by buying and selling a share at most twice

- Design a data structure that supports insert, delete, search and getRandom in constant time
- Print missing elements that lie in range 0 – 99
- Iterative Merge Sort
- Group multiple occurrence of array elements ordered by first occurrence
- Given a sorted and rotated array, find if there is a pair with a given sum

Tags: Divide and Conquer

Writing code in comment? Please use **ideone.com** and share the link here.

Anton Lisovenko · 2 months ago

Very elegant solution, though pretty diffucult to come to it independently...

mknarayan1711 · 2 months ago

How do we know which n/2 * n/2 subsquare has a missing Tile when we are trying to place a L shaped subsqaure

Ok got it the position of the missing cell is mentioned

Someone · 2 months ago

Ah, interesting solution ! Mine is less astute.

Consider a square area whose side is of size 2^k. We separate this area in four areas of size 2^(k-1). We want to fill 3 of these 4 areas (drawing a 'L'). We will call this shape a L of size k. We know how to draw a L of size 1 (this is the base shape). Suppose we know how to draw a L of size (k-1). We have a (unique) way of putting together four L of size (k-1) to draw a L of size k.

So, to fill the board, we separate it in 4 squares, we fill the 3 squares not containing the black cell and and drawingwe continue with the remaining square.

Now, we can generalize...

•

-
-
-
    - Interview Experiences
    - Advanced Data Structures
    - Dynamic Programming
    - Greedy Algorithms
    - Backtracking
    - Pattern Searching
    - Divide & Conquer
    - Mathematical Algorithms
    - Recursion
    - Geometric Algorithms
-

- ## Popular Posts

    - All permutations of a given string
    - Memory Layout of C Programs
    - Understanding "extern" keyword in C
    - Median of two sorted arrays
    - Tree traversal without recursion and without stack!
    - Structure Member Alignment, Padding and Data Packing
    - Intersection point of two Linked Lists
    - Lowest Common Ancestor in a BST.
    - Check if a binary tree is BST or not
    - Sorted Linked List to Balanced BST
- Follow @GeeksforGeeks

- ## Recent Comments

    - Ashish Aggarwal

      Try Data Structures and Algorithms Made Easy -...

      Algorithms · 17 minutes ago
    - Vlad

      Thanks. Very interesting lectures.

      Expected Number of Trials until Success · 1 hour ago
    - cfh

      My implementation which prints the index of the...

      Longest Even Length Substring such that Sum of First and Second Half is same · 1 hour ago
    - Gaurav pruthi

forgot to see that part ;)

Bloomberg Interview | Set 1 (Phone Interview) · 1 hour ago

- saeid aslami

  thanks

  Greedy Algorithms | Set 7 (Dijkstra's shortest path algorithm) · 1 hour ago

- Cracker

  Implementation:...

  Implement Stack using Queues · 2 hours ago

-