Algorithm design strategies: divide and conquer

1. Introduction
2. An application: tiling a grid
3. Data structures

1. Introduction

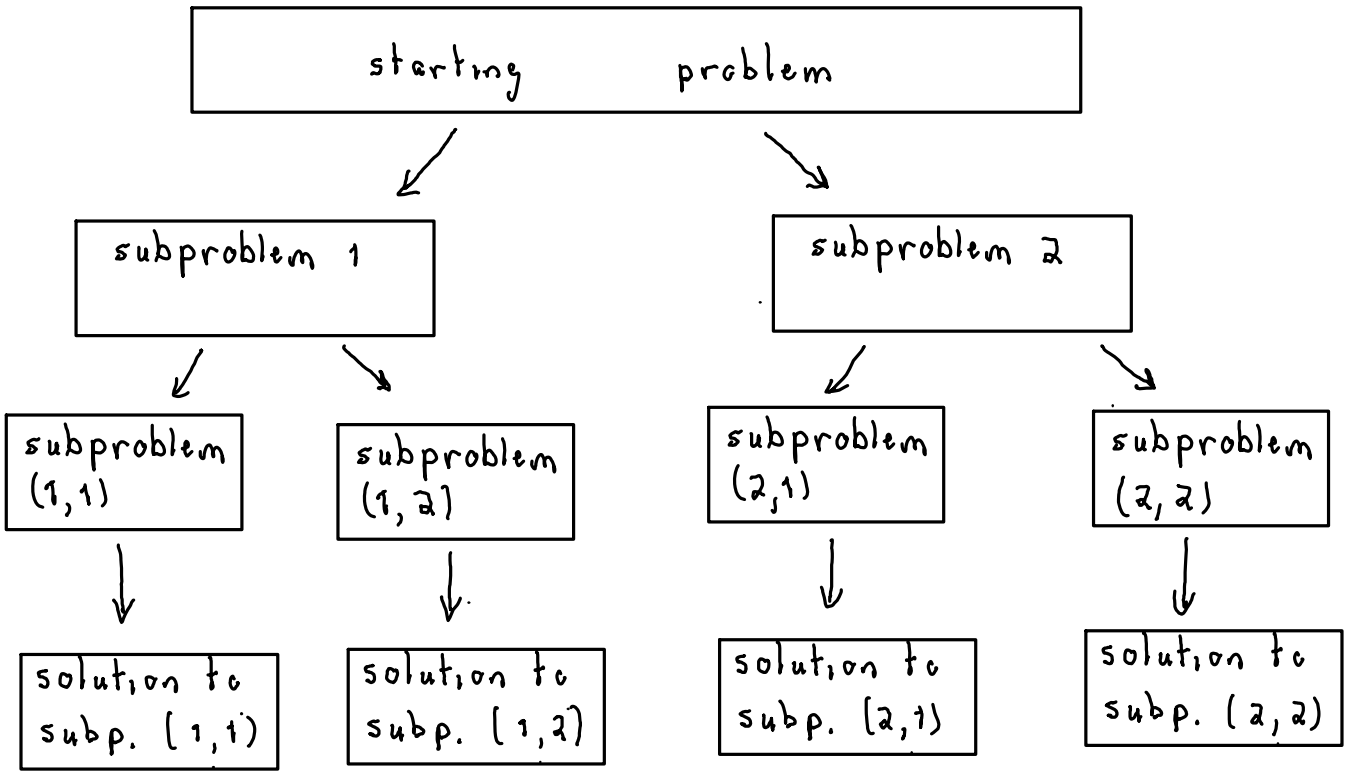
How does divide and conquer (D&C) work?

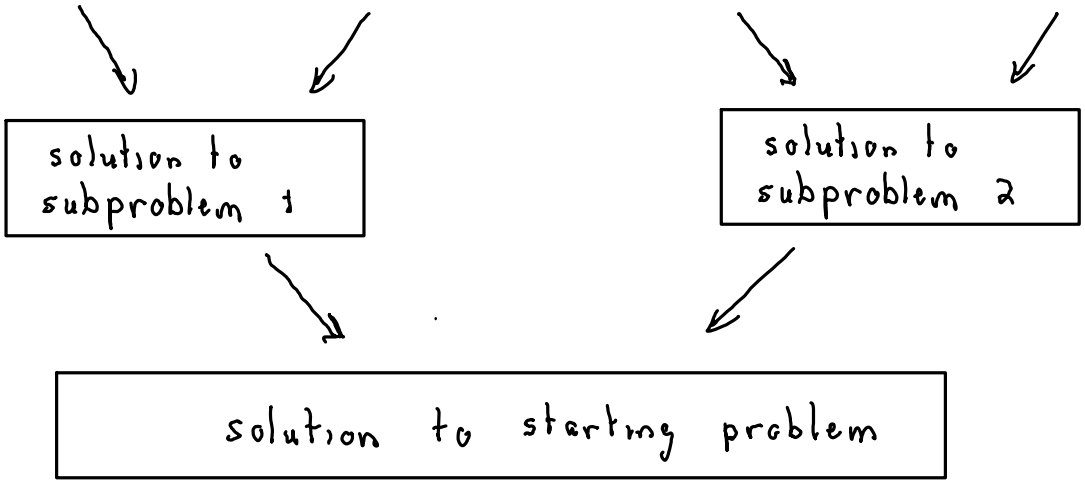
1. Divide the problem into subproblems:

• subproblems should be roughly the same size
• subproblems should be smaller versions of original problem

2. Continue dividing until the subproblem size is small enough to have a simple solution.

3. Combine the solutions of the subproblems.





Connection to recursion:

- subproblems should be smaller versions of original problem (recursion case)
- stop dividing when subproblem has simple solution (base case)

However...

An algorithm using D&C does not have to use recursion.

A recursive algorithm does not necessarily use D&C.

Why use D&C?

- nature of the problem being handled
- more efficient algorithms (possibly)
- data structure is intentionally D&C friendly

2. An application: tiling a grid

Situation:

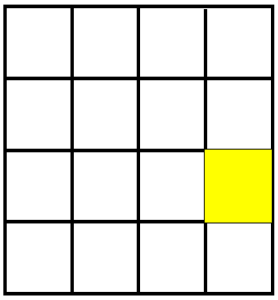
• there is a 2^n ×2^n grid of squares

• one square is already filled

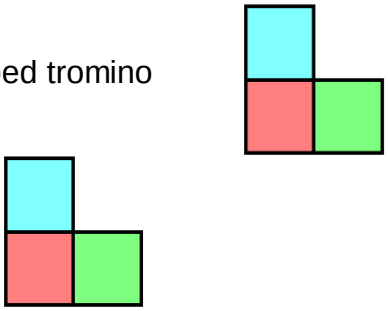
• sufficient supply of L-shaped trominoes to fill entire grid

Problem: How to place trominoes to fill entire grid?

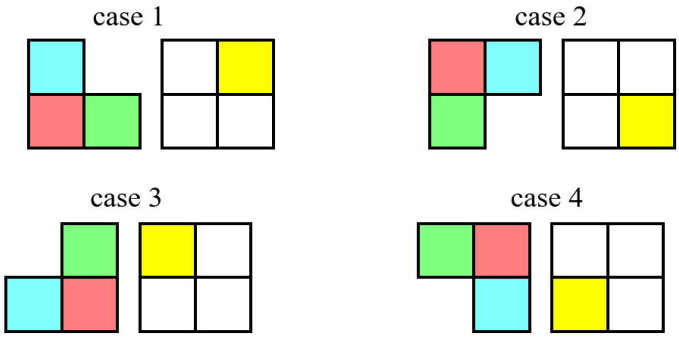
4x4 grid



L-shaped tromino

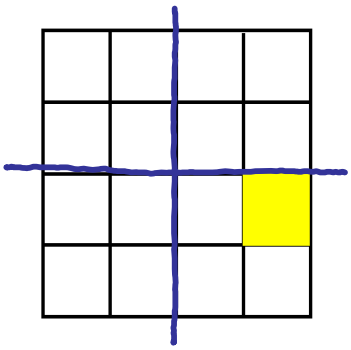


When n=1

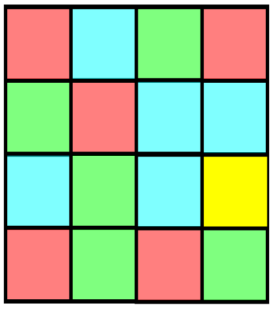


One filled grid for n=2

at start



at end

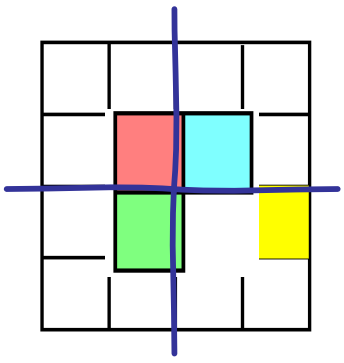


Divide and conquer:

Divide : $2^n \times 2^n$ grid can be divided into four $2^{n-1} \times 2^{n-1}$ grids (call them *quadrants*)

Conquer : fill in each of the quadrants with trominoes

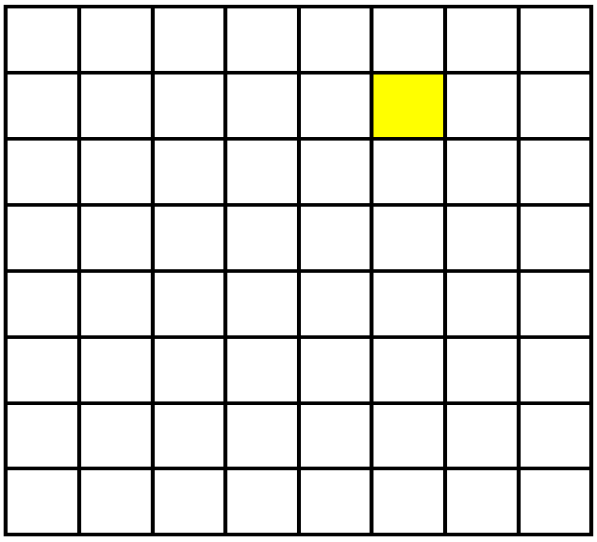
Difficulty: One quadrant is different, since it has one filled square. How can we use a tromino to make all quadrants have this same property?



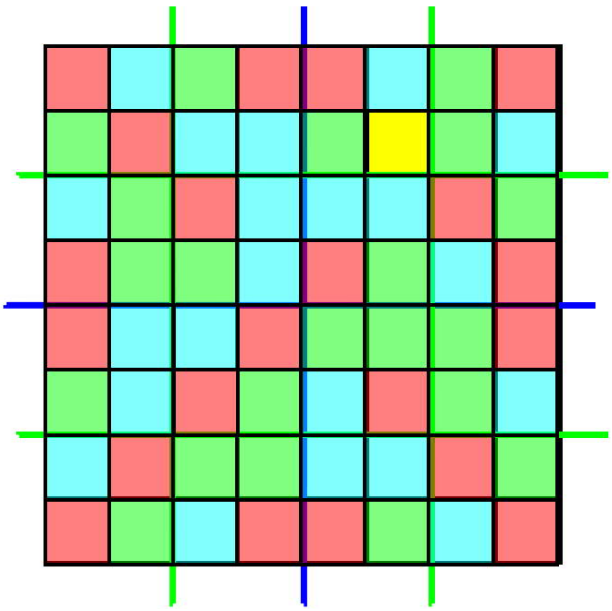
```
1   TROMINOFILL(n, loc)
2   input positive integer n, loc gives the coordinates of one filled
3   square
4   /* We fill in a 2^n × 2^n grid with L shaped trominoes. In this grid
5   exactly one square has already been filled. */
6   if n == 1 then
7      place a tromino such that it fills the empty squares
8   else
9      divide grid into four 2^(n-1) × 2^(n-1) quadrants
10     call quadrant with one filled square quadrant i
11     place one tromino at center of 2^n × 2^n grid such all quadrants
          except i have one tromino square
12     for-each quadrant
13        set locQ to be coordinates of filled square
14        fill in quadrant using TROMINOFILL(n-1, locQ)
15     end
16  end
```

Example

at start



at end



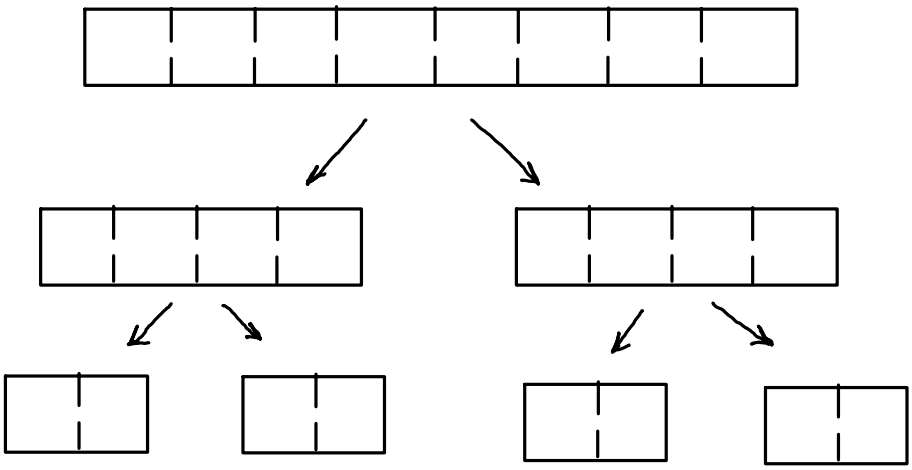
□

3. Data structures

How suitable is a data structure for D&C?

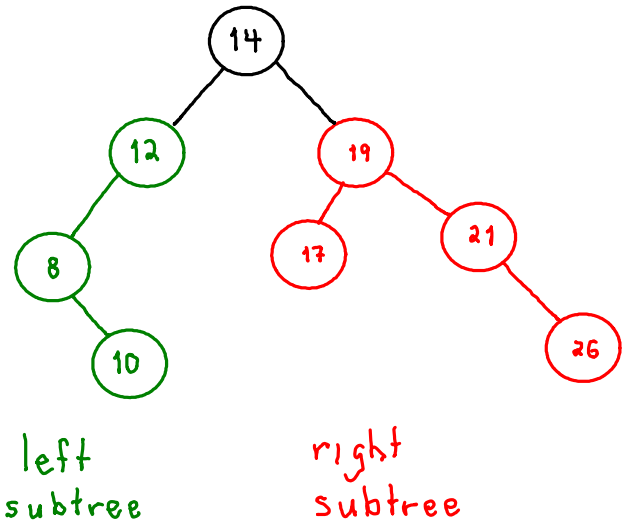
Array: a part of an array is still an array

- D&C is used in both merge-sort and quicksort for sorting fixed set of elements
- divide is used in binary search, if array is in sorted
- cannot be used for inserting new element or deleting old element



Binary search tree (BST): both the left and the right subtrees of a node are BSTs

- D&C used in traversing BST to get sorted list (inorder-traversal)
- D&C used in computing height of BST
- divide used in inserting new element, deleting old element, searching for an element


14
12
19
8
17
21
10
26
left
subtree
right
subtree