



CS 141: Intermediate Data Structures and Algorithms

Discussion - Week 5, Winter 2018



Divide and Conquer

- General idea
- Examples
- Group activities



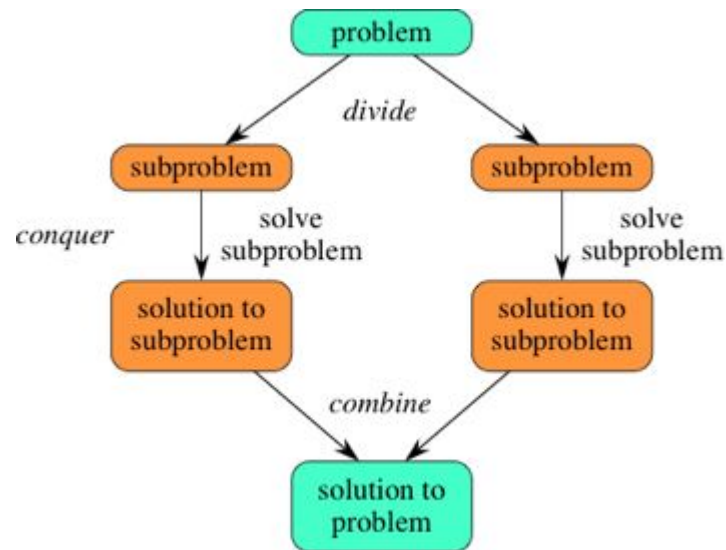
General idea

You should think of a divide-and-conquer algorithm as having three parts:

1. **Divide** the problem into a number of subproblems that are smaller instances of the same problem.
2. **Conquer** the subproblems by solving them recursively. If they are small enough, solve the subproblems as base cases.
3. **Combine** the solutions to the subproblems into the solution for the original problem.

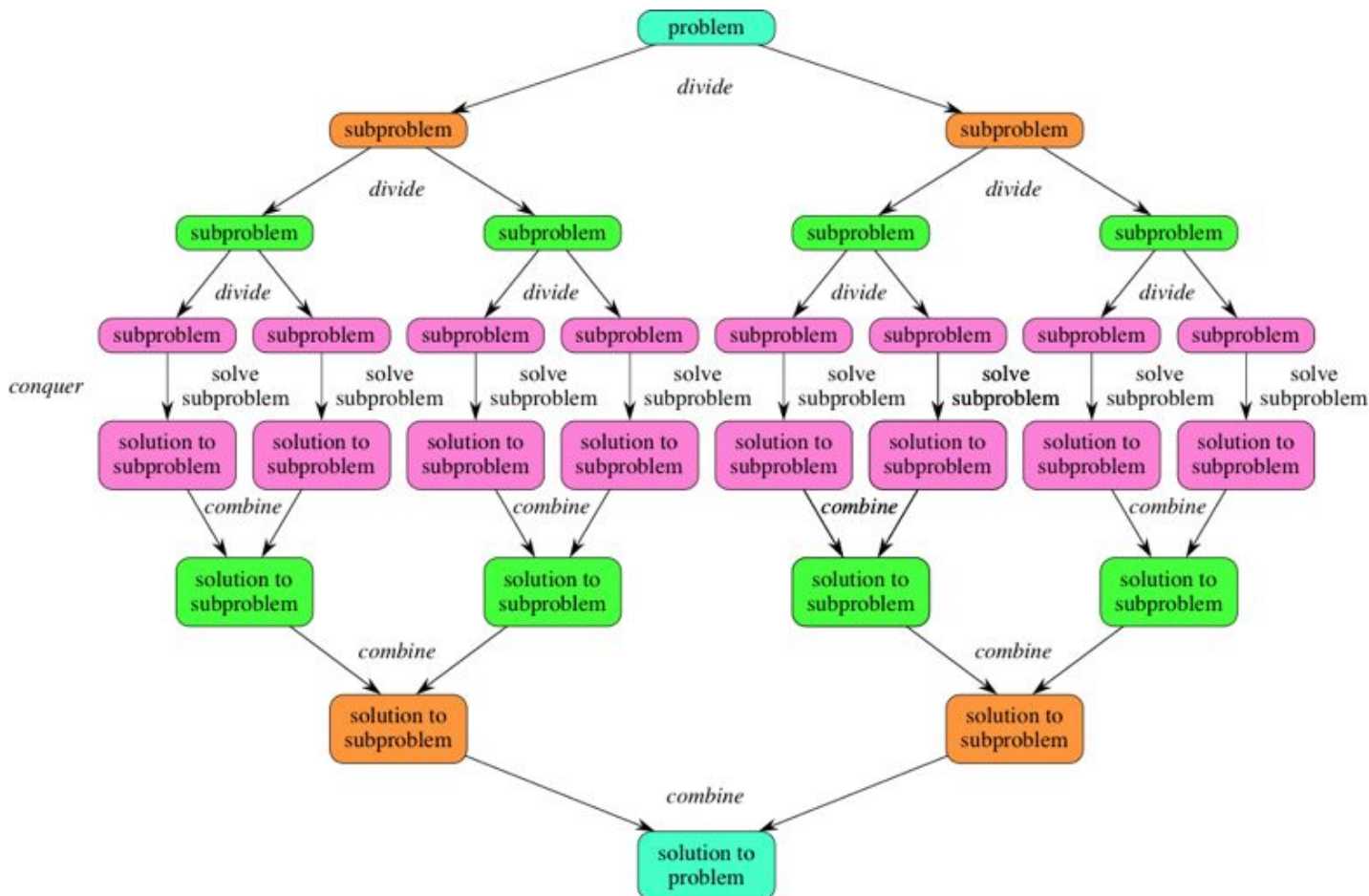


General idea



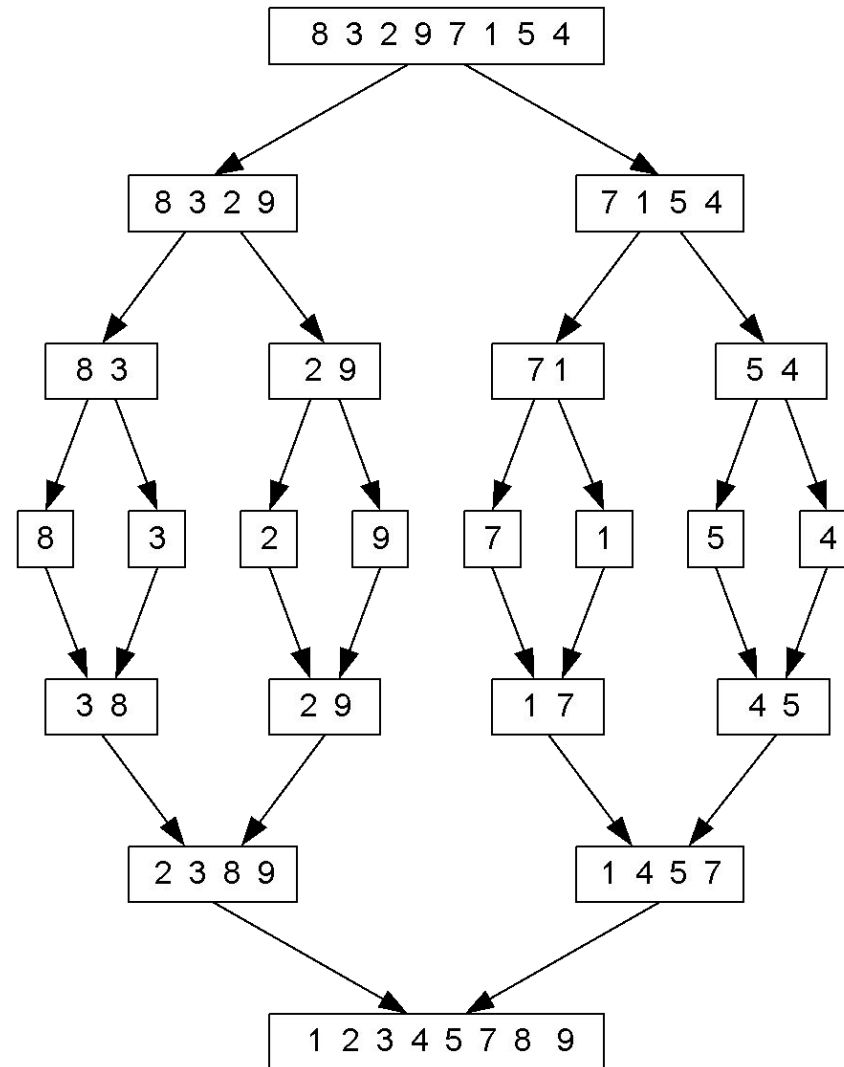


General idea



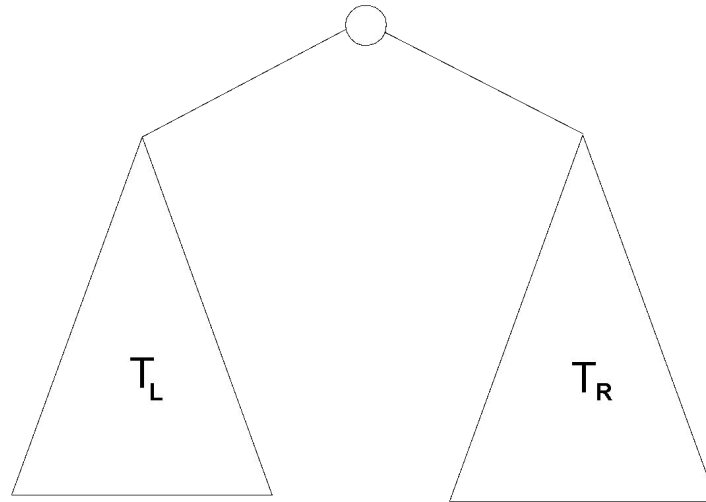


Example 1: Merge sort





Example 2: Count full nodes in a Binary tree

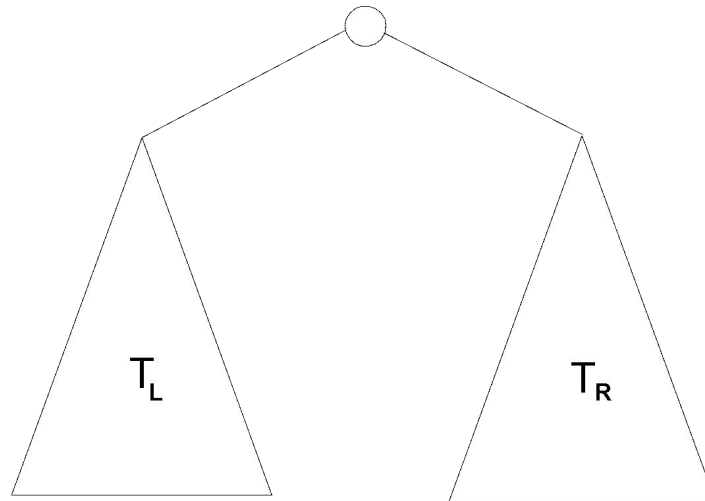


$$\mathit{count}(T) = 1 + \mathit{count}(T_L) + \mathit{count}(T_R) \text{ if } T \neq \emptyset$$

Efficiency: $\Theta(n)$. Why?



Example 3: Compute the height of a binary tree



$$h(T) = \max\{h(T_L), h(T_R)\} + 1 \text{ if } T \neq \emptyset \text{ and } h(\emptyset) = -1$$

Efficiency: $\Theta(n)$.



How to solve a given problem with D&C?

- Look at sub-problems.
- The base case for the recursion are subproblems of constant size.
- Figure out how to combine sub-solutions.
- Analysis can be done using recurrence equations or Master Theorem.



Review assignment #2