

Syllabus

7. Paradigms

- Divide and Conquer
- Dynamic Programming
- Greedy
- Backtracking
- Branch and Bound

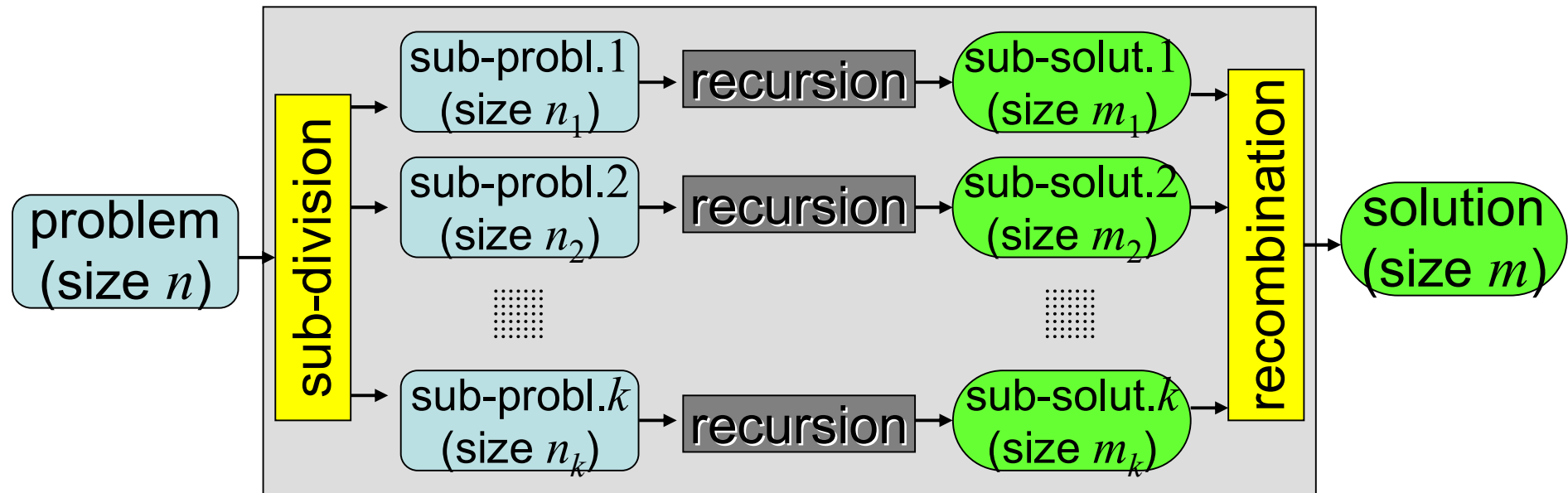
7. Paradigms

Divide and Conquer

Examples: Repeated Squaring / Fibonacci

Karatsuba, Toom, Cook, ...

Matrix Multiplication



$$T(n) = S(n) + T(n_1) + \dots + T(n_k) + R(m)$$

7. Paradigms

Dynamic Programming

Longest Common *Substring* Algorithm:

Fill table $LCS[i,j] :=$ length of longest common suffix

shared by initial segments $v[0..i-1]$ and $w[0..j-1]$

Goal:

$LCS[n,m]$

$$LCS[0,j] = 0 \quad LCS[i+1,j+1] = LCS[i,j]+1 \quad \text{if } v[i]=w[j]$$

$$LCS[i,0] = 0 \quad = 0 \quad \text{if } v[i] \neq w[j]$$

Wagner-Fischer Algorithm:

Goal: $d[n,m]$

Fill table $d[i,j] :=$ edit distance of $v[0..i-1]$ and $w[0..j-1]$

$$d[0,j] = j \quad d[i+1,j+1] = d[i,j] \quad : v[i]=w[j]$$

$$d[i,0] = i \quad = \min \{ d[i,j+1]+1, d[i+1,j]+1 \} \quad : v[i] \neq w[j]$$

Decompose the problem into *overlapping* sub-problems such that their *optimal* solutions *combine* to the original problem.

7. Paradigms

Example: *Lightest Path* from s to t in given weighted graph G

Heuristic: Repeatedly follow "cheapest" edge until arriving.

This may *not* yield the lightest path!

Cashier's Algorithm to Change-Making Problem: Express any given amount using a least number of coins/bills of values $1\text{¢}, 2\text{¢}, 5\text{¢}, 10\text{¢}, 20\text{¢}, 50\text{¢}, 1\text{€}, 2\text{€}, 5\text{€}, 10\text{€}, 20\text{€}, 50\text{€}$

Huffman Problem: Minimize expected length $\sum_{s \in \Sigma} d(l_s) \cdot f_s$

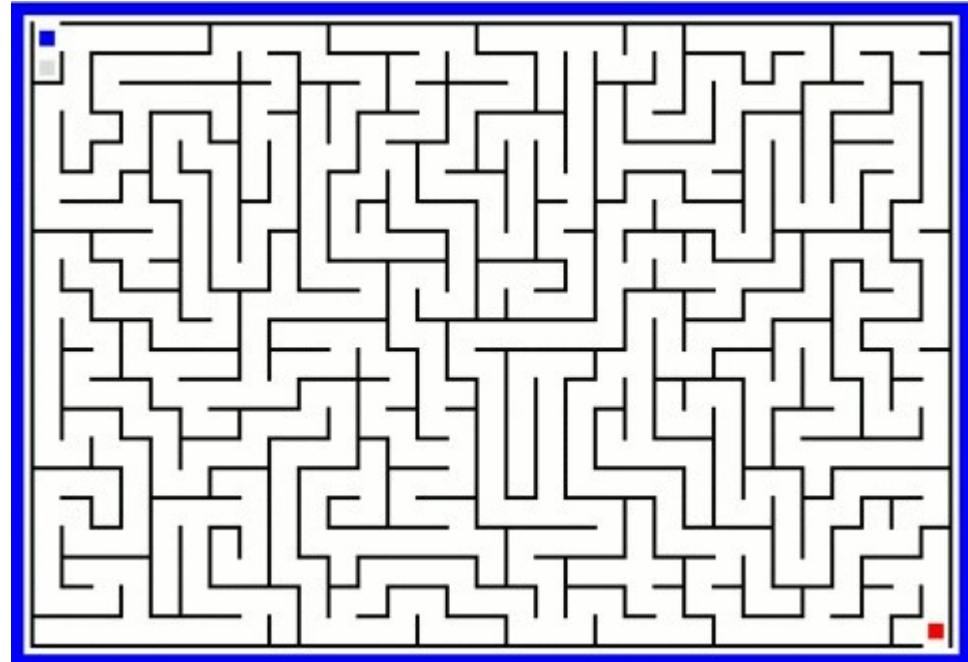
Repeatedly extract symbols $s, t \in \Sigma$ with "least" frequencies f_s, f_t .

Make whatever choice seems "best" at the moment and proceed to solve the subproblems that arise later without reconsidering previous choices.

7. Paradigms

Backtracking

Backtracking incrementally builds candidates to solutions, and abandons a candidate ("backtracks") as soon as it determines that the candidate cannot possibly be completed to a valid solution.



*Make whatever choice seems **best** at the moment and proceed to solve the subproblems that arise later **without** reconsidering previous choices.*

7. Paradigms

Branch and Bound

Definition: A tree with weighted nodes is *heap ordered* if the weight of any node is no less than that of its parent.

Problem: Find *lightest leaf* in a given heap ordered tree.

Example Algorithm:

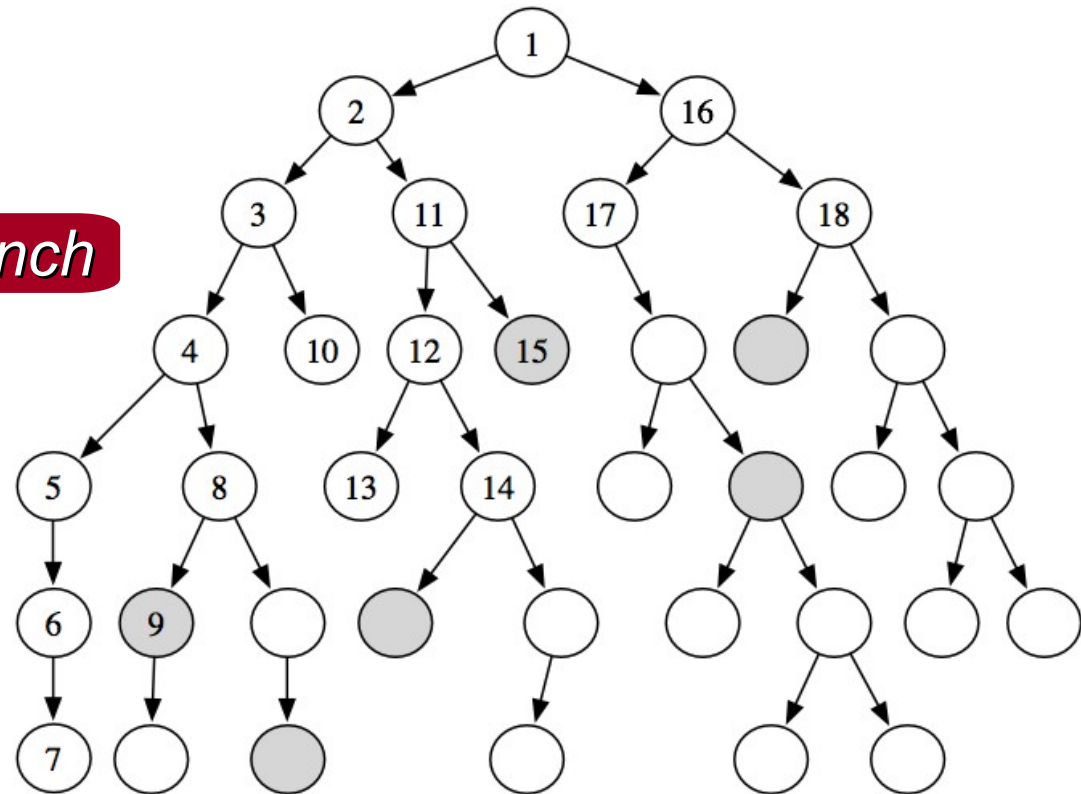
Recursively traverse the tree,

branch

keeping track of the current lightest leaf.

Refrain from recursing into subtrees whose root exceeds that weight.

bound



Summary

7. Paradigms

- Divide and Conquer
- Dynamic Programming
- Greedy
- Backtracking
- Branch and Bound