„Introduction to Algorithms"

Syllabus

6. Paradigms

– Divide and Conquer

– Dynamic Programming

– Greedy

– Backtracking

– Branch and Bound

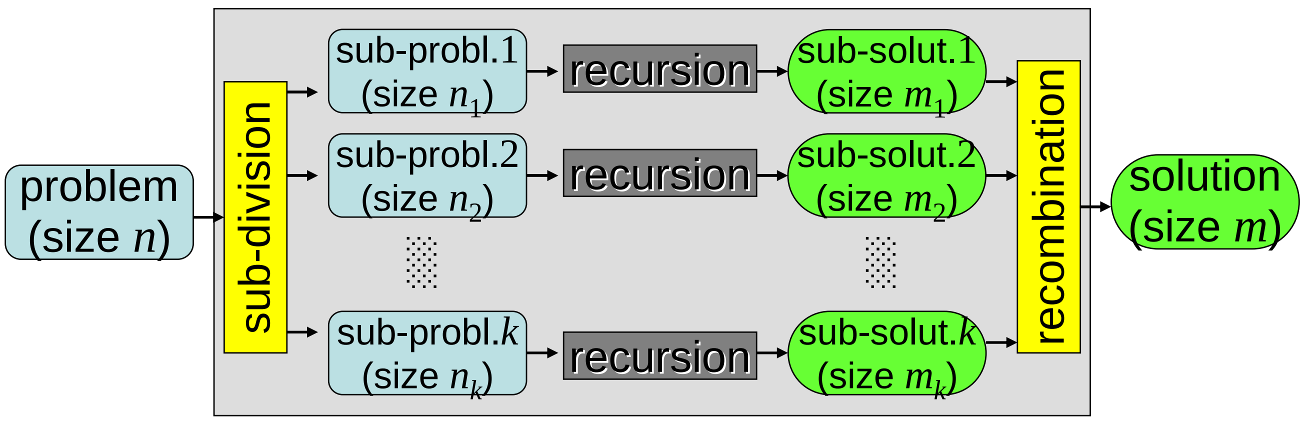
6. Paradigms

Divide and Conquer

Examples: Repeated Squaring / Fibonacci

Karatsuba, Toom, Cook, …

Matrix Multiplication



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

6. Paradigms — Dynamic Programming

Longest Common *Substring* Algorithm:

Goal: LCS[n,m]

Fill table LCS[i,j] := length of longest common <u>suffix</u> shared by <u>initial</u> segments $v[0...i-1]$ and $w[0...j-1]$

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

Wagner-Fischer Algorithm:

Goal: $d[n,m]$

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

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

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

6. Paradigms — Greedy

Example: *Lightest Path* from s to t in given <u>weighted</u> 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¢, 2¢, 5¢, 10¢, 20¢, 50¢, 1€, 2€, 5€, 10€, 20€, 50€

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

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

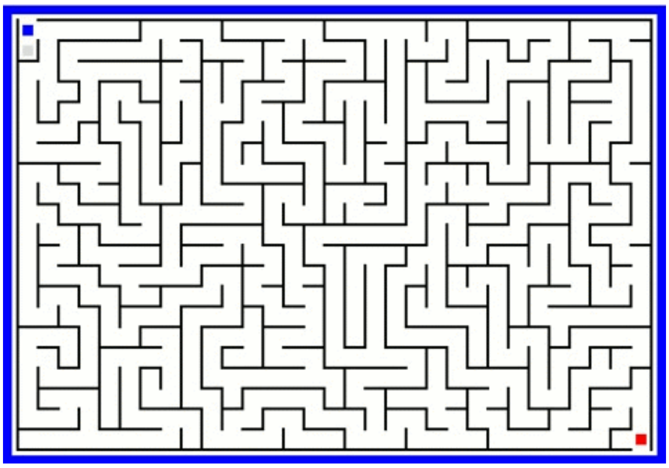
Make whatever choice seems best ***at the moment*** *and proceed to solve the subproblems that arise later with**out** reconsidering previous choices.*

6. 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 with**out** reconsidering previous choices.*

6. 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