

Dynamic programming and greedy algorithms

Lecture outline

Pierre Senellart

11 October 2018

Call the roll. Preliminary questions?

1 (From Last Lecture) Fast Polynomial Multiplication, Discrete Fourier Transform (30)

- Recall problem, decomposition, high-level algorithm scheme, complexity
- Inverse DFT, Vandermonde matrix inverse, reuse of fast Fourier transform

2 Dynamic programming (15)

- Introductory example: number of ways to make change
 - Brute force
 - Naive recursive definition
 - Asymmetric recursive definition
 - Recursive implementation, complexity
 - Memoization, complexity
 - Dynamic programming, complexity
- Modifying the problem: change with minimum number of coins
- Two main ideas: optimality of substructures, repeated subproblems
- Example case where DP applies (shortest path, cf. Floyd–Warshall) and where it does not (longest simple path)
- Theory: slides on recursion, memoization, dynamic programming
- Note on how to return the solution
- Application: 0–1 knapsack, unbounded knapsack, complexity

3 Greedy algorithms (16)

- Example: fractional knapsack, complexity
- Two main ideas: optimality of substructures, optimality of greedy choice
- Example case where greedy applies (shortest path, cf. Dijkstra) and where it does not (longest simple path)
- Back to the complexity of fractional vs 0-1 knapsack
- Basics of NP-completeness, reduction of Partition to Knapsack, implications on the notion of pseudo-polynomial algorithm