Contest programming
Dynamic programming

Maksym Planeta

27.04.2018
Table of Contents

Organisation

Introduction

Complete Search

Divide and Conquer

Dynamic Programming

Greedy Algorithms

Practice
Outline

1. Introduction
2. Linear data structures. Long arithmetic.
5. Computational geometry. Floating point arithmetic.
7. Algorithms on graphs
8. Algorithms on graphs
9. Practice session
10. Contest
Problem Solving Paradigms

- Brute Force or Complete Search
- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
Complete Search

- The simplest paradigm
- Iterate through the domain of possible solutions
- Test if a solution is correct
- Often very inefficient!
- Hint: std :: next_permutation
Example: Minimum element

- Unsorted array
- Find a minimum element
- Iterate through the whole array
Divide and conquer

- Problem can be decomposed into two independent subproblems.
- Solving independent subproblems
Dynamic Programming: Example [1]

11450 – Wedding Shopping

Given different options for each garment and a certain limited budget, buy at least one of each garment, maximizing the total cost.

Input

20 3
3 6 4 8
2 5 10
3 5 3 5

Collections

▶ 8 + 10 + 1
▶ 6 + 10 + 3
▶ 4 + 5 + 1
Dynamic Programming

- Prerequisites
  - Optimal sub-structures
  - The problem has overlapping sub-problems
Top-Down vs Bottom-Up

Top-Down
- Memo table
- Recursive search

Bottom-up
- Tabular method
- Iterative search
Greedy Algorithms

- Make a choice that *looks* optimal
- Never track back
- Faster than any other method
- May not be optimal
Greedy Algorithms: Coin change

- $c \in \{10, 1, 25, 5\}$
- Minimum number of coins to represent a given amount?
- Sort coins
- Use greedy algorithm
- Not optimal for $c \in \{4, 3, 1\}$
Task
You are given a sequence of $n$ integers $a_1, a_2, \ldots, a_n$ in non-decreasing order. In addition to that, you are given several queries consisting of indices $i$ and $j$ ($1 \leq i \leq j \leq n$). For each query, determine the most frequent value among the integers $a_i, \ldots, a_j$. 
11235 – Frequent Values

Solution

- Naïve algorithm
- Segment tree
- What to store in the leafs?
Practice

Solve following set of problems in a group:

1. 00787 – Maximum Sub-sequence Product
2. 11264 – Coin Collector
3. 11292 – Dragon of Loowater
4. 11413 – Fill the Containers
5. 00624 – CD
Home reading

Cormen.

1. Recommended
   ▶ Section 15. Dynamic Programming