Common instructions: Develop an algorithm. If you like, try on a computer. Describe your solution in a documentation containing the problem definition, your algorithm, its analysis as to the algorithmic complexity and main implementation pros and contras, results on several data sets. Send me the documentation in a pdf file.

Incremental and D&C algorithms – problems to solve

Star polygonization

Construct a star polygonization of the given set of points S (i.e., a star-shaped polygon whose vertices are the given points, see lecture 2) by a) incremental insertion, b) incremental construction; more solutions may exist.

MergeSort

Sort the given set of integer values by Mergesort using D&C principle., see lecture 3.

ChessBoard tiling

A chessboard of the size $2^n x 2^n$ is given with one empty square, tile it by a T-shape tiles by an algorithm based on D&C so that the only uncovered square if the empty one. See lecture 3.

Dynamic programming – problems to solve

The knapsack problem

There are N geometric objects with masses m_1, …, m_N and a mass M (capacity of the knapsack). The task is to choose some objects in such a way that the sum of their mass is as big as possible, but under M. Solve with time complexity *O(MN).*

Ex.: N=3, m={2,6,7}, M=10. We will get maximally the mass 9 by m_1 and m_3.

Control question (after you find the algorithm): What would happen if we filled the auxiliary array in the opposite direction?

Shortest paths (Floyd-Warshall algorithm)

We have N towns. There are bi-directional roads between some pairs of the towns, the lengths are given as input. We suppose that the roads do not meet elsewhere than in the towns (if they cross, then as fly-overs). Compute the shortest distances (the lengths of the shortest paths) between all pairs of towns. Solve in $O(N^3)$.

Control questions:

1. Find not only the lengths, but also the indices of towns on these paths.
2. How to modify if the roads are one-directional?
3. How can we know that if we join two paths, we get again a path (i.e., that there are no duplicities)?
4. Mind the order of loops in the algorithm.

Coin change

Find the number of ways how to make change for a particular amount of cents, n, using an infinite supply of each of $S=\{S_1,S_2,\ldots,S_m\}$ valued coins? The order does not matter.

Ex.: n=4, S={1,2,3}, there are 4 solutions:{1,1,1,1},{1,1,2},{2,2},{1,3}

Control question:

1. How does the algorithm change if we want also to know the solutions, not only their number?

A maximum partial sum in a sequence

Let us have a sequence of real numbers $r_1,..,r_n$. Find $M=\max S_{ij}$ for $1<=i<=j<=n$ where $S_{ij} = r_i + r_{i+1} +\ldots + r_j$, $i<j$, $S_{ii}=r_i$ for i=j.

Ex.:

j	1	2	3	4	5	6	7
rj	1	-5	3	-1	2	-8	3
Mj	1	-4	3	2	4	-4	3

M=4

Minimum Weight Triangulation

A triangulation of a polygon P={v_1,v_2,..,v_n,v_1} is a set of non-intersecting diagonals that partition the polygon into triangles. We say that the weight of a triangulation is the sum of the lengths of its diagonals. Find the minimum weight triangulation for a given polygon P.

Backtracking – problems to solve

Parentheses

Generate all correct arrangements of N pairs of parentheses.

Ex.: N=3, ()()(), (())(), ()(()), (()()),((()))

Directory search

Make an algorithm which searches through a given tree of directory, to find the first file/all instances of a file of the given name.