2. Problem set on

Algorithms, Data Structures and Data Abstraction WS 18/19

Wolfgang Mulzer

Due on 02. November 2018, 10:00, in the respective TA's mailbox

Problem 1 Sparse Polynomials *10 Points*

In class, you saw two ways to represent a polynomial of degree n. The coefficient representation becomes inefficient if many coefficients of the polynomial are 0 (such polynomials are called *sparse*). Develop an efficient representation for sparse polynomials and describe algorithms for addition, multiplication, and evaluation in this representation. Analyse the efficiency of your algorithms in dependence of the degree n of the polynomials and the number k of nonzero coefficients.

Extra credit (5 Points): Implement your data structure in Java or Python.

Problem 2 Sorting Experiments *10 Points*

In "Imperative and Object Oriented Programming", you saw several algorithms for sorting an array of n numbers. Choose three of them, and implement them in Java or Python. Among your algorithms, there should be one with worst-case running time $\Theta(n^2)$ and one with worst-case running time $\Theta(n \log n)$. Test your implementations with random inputs and describe for each algorithm a class of inputs that represents an "extreme" case.

Measure the number of milliseconds that your implementations need for inputs of at least one hundred different sizes. For each input size, pick at least ten random inputs.

Discuss your results. What are the advantages and disadvantages of the respective algorithms? Which algorithm would you use under which circumstances? What do your experiments tell you about the O-Notation?

Problem 3 The Towers of Hollywood *10 Points*

You probably know the legend of the towers of Hanoi that was invented by the French mathematician Édouard Lucas in order to sell a wooden puzzle.

In the great temple of Benares, there is a wooden plate on which there are three vertical rods made out of diamond. On these rods, there are n disks made of pure gold, of n different sizes. At the beginning of time, the god Brahma has put the disks onto the first rod, sorted according to their size, such that the largest disk lies at the bottom. The Brahmins in the temple work tirelessly and transport the disks from rod to rod, obeying the following rules: (i) in each step, at most one disk can be moved from one rod to another; and (ii) a larger disk may never lie on top of a smaller disk. As soon as all the disks have reached the final rod, the world will end.

(a) Describe a recursive algorithm that allows the Brahmins to reach the end of the world quickly. Determine exactly the number of moves that are necessary for n disks. How many years do we need to wait for the end of the world, assuming that Brahma dropped 100 disks onto the first rod and that it takes one day to move a single disk?

(b) Recently, the temple has moved to California, where the Brahmins have adopted a much more relaxed attitude. The disks are now made out of gilded cardboard, and the rods are plastic. Most importantly, the rules for the disk ordering have been relaxed: we still require that on each rod the largest disk lie at the bottom, but the order of the other disks is arbitrary. In the end, all disks again must reach the last rod in the correct order, largest to smallest from bottom to top.

Describe an efficient algorithm for the relaxed problem. Determine the exact number of moves your algorithm requires for n disks. How many years do we need to wait now for the end of the world, assuming that Brahma dropped 100 disks onto the first rod and that it takes one day to move a single disk?

Credits: This problem is due to Jeff Erickson.
`http://www.cs.uiuc.edu/~jeffe/teaching/algorithms/hwex/s09/hw0.pdf`