Data Structures and Algorithms **Spring-Summer 2011/12**

Assignment 6

**Valeria Fionda, Mouna Kacimi,
Werner Nutt, Simon Razniewski**

Lists and Array Representation of Objects

Instructions: Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
Be prepared to present your solution at the lab. If you are not able to explain your solution, this will be considered as if you had not done your work at all.
You can write up your answers by hand (provided your handwriting is legible) or use a word processing system like Latex or Word. Experience shows that Word is in general difficult to use for this kind of task.
For a programming task, your solution must contain (i) an explanation of your solution to the problem, (ii) the Java code, in a form that we can run it, (iii) instructions how to run it. Also put the source code into your solution document. For all programming tasks, it is not allowed to use any external libraries ("import") or advanced built-in API functions (for example, `String.indexof("a")` or `String.substring(1,5)`), if not stated otherwise.
Please, include name, matriculation number and email address in your submission.

1. Manipulating Polynomials

Many applications require to manipulate polynomials of arbitrary order. For this exercise, you are asked to solve the following tasks:

1. Develop an algorithm that *adds* two polynomials and write it down using a pseudo-code.

   Represent a polynomial by a linked list structure, where each node holds information about a single term of the polynomial. An example of addition of two polynomials is given in the following:

   1^{st} *Polynomial*: $3x^{14} + 5x^{11} + 4x^9 + 9x^8 + x^3$

   2^{nd} *Polynomial*: $8x^8 + 2x^7 + 3x^6 + 6x^3 + 7x^2 + x + 12$

   Result: $3x^{14} + 5x^{11} + 4x^9 + 17x^8 + 2x^7 + 3x^6 + 7x^3 + 7x^2 + x + 12$

2. Develop an algorithm that *multiplies* two polynomials and write it down using a pseudo-code.

   As before, represent each polynomial using a linked list data structure. Recall hat two polynomials are multiplied as follows:

$$\sum_i a_i x^i \times \sum_j b_j x^j = \sum_{i,j} a_i b_j x^{i+j}$$

3. Analyze the running time of the two algorithms. Justify your answer.

4. For both, addition and multiplication, develop algorithms that use an array structure where a slot is kept for each possible exponent of x. For example, the array representation of the first polynomial in the example above is:

3	0	0	5	0	4	9	0	0	0	0	1	0	0	0

5. Implement the two sets of algorithms in Java.

6. For each operation, compare the algorithm over linked lists with the one over arrays. Which one is faster and which one uses less memory?

Instructions: When working on the exercise, please take into account the following instructions.

- For each algorithm, choose the list version that is most appropriate (linked list, linked list with head and tail, doubly linked list, etc.) and explain your choice.

- For the running time experiments, generate sparse high order polynomials. For example: a polynomial of order 100 which is (6%) sparse means that the highest exponent of x is 100 and there are only 6 terms. To obtain a realistic picture, generate several inputs for each order and sparsity degree and measure how long the algorithm runs on them. Gradually increase the order and the sparsity and run the experiment for each of these values.

- For each comparison, make sure you measure the running time for the same input.

- State your observations and discuss possible reasons for them.

Hint: Consider increasing the memory size if needed, using virtual machine option `-XX:AggressiveHeap`.

(30 Points)

Submission: Wednesday, 2 May 2012, 8:30 am. Preferably by email.

If you want to submit a hand-written solution, hand it over to your lab tutor or in the lecture. However, all code has to be submitted in electronic form.