Computer Assignment

IBC016 Combinatorics

October 6, 2017

Obligatory

Note that it is a prerequisite for passing the course that you complete this assignment successfully. In particular this means that you score at least a 5.5.

In addition, the grade that you get for this assignment will be included as a ninth homework assignment and it will increase the bonus you get for your homework.

This assignment is not obligatory for students who already did the computer assignment in previous years. Please contact Engelbert to inform him about such a situation explicitly.

Groups

This work should be done in groups of three students. Only students who failed the course in earlier years are allowed to do this assignment on their own.

Please register your group in Blackboard by enrolling three students into it. The option to register for the groups for this Computer Assignment will become available on Saturday October 7, 08.00 am.

Goal

The goal of this assignment is that you will become experts on the algorithm for deriving direct formulas given a recurrence relation. The last few years students complained in the evaluations that this topic was treated to fast in the course, so now we give it a lot of attention, so everybody is prepared when this algorithm needs to be applied at the exam.

Task

The assignment consists of the following parts:

- Write an implementation of the algorithm mentioned above.
  - You may use your favorite programming language.
  - You may use existing libraries.
  - However, your program may not be based upon Computer Algebra systems like `Maple`, `Mathematica`, `Sage` or `WolframAlpha`.
  - Your program should be able to read a `.txt` file using the input format as shown in the examples below.

  - These files will have the name `comass??.txt` where there are digits on the place of the question marks.
  - Your program should be able to write the solutions to a `.txt` file in the output format as shown below.
  - Please use the name `comas??-dir.txt` where the question marks are replaced by the proper digits.
  - Ideally, your program gives exact solutions, that is, including fractions and square roots. However, if that is too hard to implement, you may use numeric approximations, but you won't be able to score a 10 in that case.
  - You should submit a `.zip` file of your source code into the corresponding Blackboard assignment before Friday October 27.

- On Friday October 27 you will have to prove that your program works by joining in a competition.
  - The competition will be held in one of the lecture rooms, probably LIN4.
  - Although this competition could technically be done without being present at the same spot, you are obliged to be present, simply because we want to be able to see your program in action. If this is impossible for you, please let Engelbert know as soon as possible.
  - We will try to bring some power extension cords, but try to make sure that your laptops have enough power left to survive 45 minutes.
  - At 13.45 a list of input files will be made available through a Blackboard assignment and you have 45 minutes to find as many solutions as possible.
  - This list of files contains a range of recurrence relations: homogeneous, nonhomogeneous, different degrees, nonhomogeneous parts that are polynomial, exponential or linear combinations of both.
  - Your program does not need to be fully automatic. For instance, if it is difficult to detect automatically what the type of the nonhomogeneous part is, but you can spot this immediately by looking at it, you may simply tell your program interactively what to do.
  - If it turns out that your program doesn't work as expected, you may even change the source code within these 45 minutes.
  - You have to hand in your final submission before the Blackboard assignment closes at 14.30. This submission should include a `.zip` file containing:
    - All `comass??-dir.txt` output files that your program managed to generate.
    - A one-page PDF report describing which language you used, which libraries you incorporated and which problems you encountered. Most of this report can already be prepared before the competition actually starts.

Grade

The grade will be determined by the correct solutions you hand in. Solutions will be checked for correctness by computing the first twenty values of the sequences and comparing the output of both the original recurrent formula and your direct formula. If the absolute difference for each of these values is less than $\frac{1}{1000}$ your solution is considered correct.

Input format

Here are some examples.

- `comass03.txt`

  ```
  eqs :=
  [
  s(n) = -4*s(n-2) + 4*s(n-1),
  s(0) = 6,
  s(1) = 8
  ];
  ```

- `comass07.txt`

  ```
  eqs :=
  [
  s(n) = s(n-1)+s(n-2),
  s(0) = 1,
  s(1) = 1
  ];
  ```

- `comass16.txt`

  ```
  eqs :=
  [
  s(n) = n^3 + 8*s(n-2)-16*s(n-4),
  s(0) = 0,
  s(1) = 1,
  s(2) = 2,
  s(3) = 3
  ];
  ```

- `comass36.txt`

  ```
  eqs :=
  [
  s(n) = -2*s(n-1)+11*s(n-2)+12*s(n-3)-36*s(n-4) +41^(n-4)+3,
  s(0) = 1,
  s(1) = 1,
  s(2) = 1,
  s(3) = 1
  ];
  ```

Some remarks:

- In this document long lines are split into smaller lines for typesetting purposes. In the files that are actually being used this will not be the case.
- You may assume that the first equation within the set is the actual recurrence relation and the remaining equations are the initial conditions.
- You may assume that the first equation is always of the form $s(n) := \ldots$.
- You may not rely on a specific amount of whitespace between terms.

Output format

And here are the expected output files for the examples given above.

- `comass03-dir.txt`

  ```
  sdir := n -> -2*2^n*(n-3);
  ```

- `comass07-dir.txt`

  ```
  sdir := n -> 1/10*(1/2*5^(1/2)+1/2)^n*5^(1/2)+1/2*(1/2*5^(1/2)
      +1/2)^n+1/2*(-1/
  2*5^(1/2)+1/2)^n-1/10*(-1/2*5^(1/2)+1/2)^n*5^(1/2);
  ```

- `comass16-dir.txt`

  ```
  sdir := n -> 139/216*(-1)^n*2^n*n+1/9*n^3+161/162*(-1)^(n+1)*2^n
      +41/8*2^n*n+16
  /9*n^2-47*2^(n-1)+32/3*n+1984/81;
  ```

- `comass36-dir.txt`

  ```
  sdir := n -> -71/1650*(-1)^n*3^n*n+5089/60500*(-1)^n*3^n
      -254/975*2^n*n+138484/
  190125*2^n+1/2944656*41^n+3/16;
  ```

Some remarks:

- Note that it is very well possible to write essentially the same direct formula in many different ways, so these are just given as an example. So don't worry if you test your program with the corresponding input files, but get seemingly different solutions. Obviously, it would be wise to check whether the first twenty values are the same or not.
- Fractions are written in the normal way.
- Square roots should be done by raising to the power (1/2), as you can see in example `comass07-dir.txt`.
- Use parentheses to enforce the proper order of evaluation of the operators.
- Please let the result be an assignment to `sdir`, because that helps in uniformly checking your solutions.

Questions

If you have any questions about this assignment, please let Engelbert know as soon as possible.