Computer Assignment

IBC018 Discrete Structures

December 21, 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.

You can obtain a bonus with it. For this course you can get the following grades: $IT1$ intermediate test 1, $IT2$ intermediate test 2, EX final exam and CA computer assignment.

Now define grade

$$G \quad := \quad \max(EX, \tfrac{2 \cdot EX + IT1 + IT2}{4})$$

then you can compute your final grade by

$$FG \quad := \quad \begin{cases} G & \text{if } G < 5 \\ \min(10, G + \frac{CA}{10}) & \text{if } G \geq 5 \end{cases}$$

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 Wednesday December 13, 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.
  - However, in Blackboard you will find a framework for reading the input files and writing the output files written in `Python`, so this might be the best choice.
  - But note that it is not obligatory to use this framework!
- 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 February 2.

- On February 2 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 HG00.068.
  - 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 number of 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.

Some answers

This assignment has been done in the Combinatorics course as well. This is the list of their questions:

1. Can we use libraries like `SymPy`?
   - Yes, but you may not use the command `rsolve`.
   - *Note that you have to implement the given algorithm. You are not allowed to use a different algorithm that you found somewhere else.*
2. Should we implement things like Gauss elimination ourselves or can we use libraries for that?
   - You may use libraries for that.
3. May we assume that the nonhomogeneous part $F(n)$ is always separated from the homogeneous part by spaces?
   - No, $F(n)$ can be anywhere in the right hand side of the first equation, with or without spaces.
4. Is the Computer Assignment obligatory?
   - Only if you want to pass the course...
5. We cannot find C/C++ libraries that solve equations of arbitrary degree. Should we write our own methods for solving characteristic equations of higher degree?
   - Maybe it is wise to start with a program that can find the roots of characteristic equations of degree two and degree three?
   - I also don't have algorithms to solve all equations of higher degree, so if you get a characteristic equation of a high degree, it is quite likely that it has some simple integer solutions in the range of -5 till 5. And if you find such a simple solution, you can decrease the degree of the original equation...
   - And did you look at `https://www.singular.uni-kl.de/Manual/4-0-3/sing_864.htm`?
6. Can we use three laptops or just one?
   - You may apply divide and conquer techniques by letting every team member use his or her own laptop.
7. Is the first initial condition always $s(0)$ or can it also be $s(3)$?
   - You may assume it is $s(0)$.
8. May we assume that the coefficient of $s(m)$ is always a positive or negative integer like $24s(n-1)$ and not some formula we have to evaluate like $(9+5\cdot 3)s(n-1)$?

- No, in `comass33.txt` it says:

  ```
  s(n) = (9/2)*s(n-2) +(3/2)*s(n-3)-5*s(n-4)-3*s(n-5)
         + (n-5)^2-3*(n-5)+7
  ```

  So the coefficients need not be integers.
- However, note that in the nonhomogeneous part it is possible that there are terms you may want to simply before processing.

9. Can it be that we have to simplify something like $s(n) = s(n-1) + 4s(n-2) + 2s(n-1)$ to $s(n) = 3s(n-1) + 4s(n-2)$?

   - No, but as you can see in `comass03.txt` it can happen that the order of $s(n-1)$, $s(n-2)$, ... is not always decreasing.

10. We are not allowed to use `rsolve` in `SymPy`, but are we allowed to use `solve`?

    - Yes.

11. Are we allowed to manually change the input file to do some preprocessing?

    - It is not the idea that this will be done, but I am afraid I won't be able to detect this when checking your output file...

12. If we don't use exact values, does this mean that the solution is automatically incorrect?

    - No. Using your output file I'll compute the first twenty values of the list and compare them with my solutions. If the absolute difference for each of these values is less than $\frac{1}{1000}$ your solution is considered correct.

13. Can we expect a nonhomogeneous part of the form $F(n) = n^{-3}$?

    - No, $F(n)$ is composed out of polynomial functions like $(n-2)^3 - 3n + 37$ and out of exponential functions like 3^n. Hence the most difficult type would be something like $F(n) = 3^n + 2^n + n^2 - 3$.

14. Should we expect recurrence relations of degree higher than 9?

    - Yes, there are some examples with degree 10. But as far as I recall, that is the maximum.