

# Orderings

Dylan Zwick

Wednesday, June 18th, 2008

Today we start on programming and implementing some of the things we've learned so far in Prof. Bertram's lectures. Namely, we're going to be writing programs that implement different monomial ordering schemes.

Now, the input/output methods for these programs will be very primitive, as we're only going to be getting our feet wet here as far as software development is concerned. The important thing is that we get some experience writing code that implements algorithms discussed in the book.

## 1 Getting Started

First, go to the webpage I've created for this class (I'll be writing the name on the board) and download the source code for `lexorder.cpp`. Save this source code to an appropriate folder in your account, and then open up the file using emacs.

We're going to step through the program in class. This program introduces some new concepts and data structures, most notably arrays. Also, this program is highly commented, unlike the programs we went over yesterday.

## 2 Your Job

I want you to use the lexicographic ordering program as your template for making two other programs similar to it. These programs can (and you'll probably want to do it this way) just directly copy and paste the

input/output format from the lexicographic order program I've written. However, instead of implementing lexicographic order, you should write two programs that implement graded lex and graded reverse lex, respectively.

When you're done writing these programs and have tested them to your satisfaction, please email me the source code that you've written. I'll compile and run it on my computer to see how it works and take a look at your code so I know how you're doing.

### **3 Looking Forward**

I was thinking that we'd probably be able to implement most of what we want to get done for this class in C++ without too much effort in the way of dealing with C++ features that do not relate directly to what we're talking about in class. I'm starting to revise that opinion. So, before Monday I'm going to be doing some research and probably writing a bunch of code that implements stuff we'll need like a polynomial class. So, the plan is that this week will be comparatively light on the computer stuff, just so everybody (including myself) can get familiar with programming and the kind of algorithms we'll need to implement. However, starting on Monday we'll have much more complicated computer programming projects to implement, with the goal that by the end of next week you'll have written algorithms that completely implement Groebner bases in a useful way. This will be the essential meat of the computational aspect of this class.