AIMS DMG Exercises 06
AIMS 2013-14: Designs, Matroids and Graphs

Rob Beezer
University of Puget Sound

Nancy Neudauer
Pacific University

January 6, 2014

Exercise 1. Consider the system of equations below. When you are asked for a solution, you can expect "nice" entries, so if you get a mess of fractions, go back and see if you entered the problem properly. This problem is *not* over a finite field (like they were in lecture). Instead use the rational numbers as your field, since Sage will do these computations exactly. This is `QQ` in Sage.

$$\begin{aligned}
x_1 - 3x_2 + 8x_3 + x_4 + 2x_6 &= 7\\
-3x_1 + 2x_2 + 4x_3 + x_4 - 7x_5 + 2x_6 &= 9\\
-x_1 - x_2 + 8x_3 + x_4 - 4x_5 + 2x_6 &= 11\\
2x_1 - 4x_2 + 8x_3 + x_4 + 2x_5 + 2x_6 &= 5\\
-x_1 + x_2 - x_4 - 2x_5 - 2x_6 &= 5
\end{aligned}$$

1. Use a coefficient matrix and a vector of constants, along with Sage's `.solve_right()` command to find a single solution to the system. While this system has infinitely many solutions, your answer should be *the one* solution Sage produces with the indicated commands, not just *any* solution you can find by any method you wish.

2. Check that your solution is correct by multiplying the solution vector by the coefficient matrix and testing equality (`==` in Sage/Python) with the vector of constants.

3. Check that your solution is correct by using the entries of the solution vector to form a linear combination of the columns of the coefficient matrix. Then test equality of this linear combination with the vector of constants.

4. Build three different solutions to the system above by adding *diverse* elements of the null space of the coefficient matrix (`.right_kernel()` in Sage).

5. Construct a new vector of constants for the system (i.e. the right-hand sides of the equations) so the system has no solution. Do not guess at such a vector, but instead, *explain* an approach that uses the column space of the coefficient matrix.

To receive full marks you should have all of the computations requested. There should be some variety in the four solutions, and the final part should contain a well-written explanation of your method, and not just a single vector.

Exercise 2. Vector spaces with a field of scalars that is infinite will typically have infinitely many bases. Sage can construct two bases for the null space of a matrix (which, of course, is a vector space). The default has zeros and ones in the first entries of the basis vectors, while the second, more natural basis, has zeros and ones in the later entries of the basis vectors. The second is obtained with the `basis='pivot'` keyword.

The commands below illustrate how to obtain these two bases. *NOTE:* Due to a bug in Sage (*mea culpa*), once you compute a right kernel of a matrix, Sage will remember it and it cannot be changed until you change the matrix. So make a copy of the matrix, compute the bases and save them in new variables before proceeding.

```
A = matrix(QQ, [
[-247, 207, 9, -692, -1155, 9, -692],
[-152, 128, 6, -426, -712, 6, -426],
[46, -39, -2, 129, 216, -2, 129],
[95, -79, -3, 266, 443, -3, 266],
[-30, 25, 1, -84, -140, 1, -84]
])
B = copy(A)
```

```
eb = A.right_kernel().basis(); eb
```

```
pb = B.right_kernel(basis='pivot').basis(); pb
```

Your goal in this problem is to “exchange” vectors to convert one basis into the other, you can choose either one to be “first”. More specifically, start with one basis and remove a vector. Now you should be able to add a single vector from the second basis to this set, and return to a basis of the null space. (This step could take some trial-and-error work since you cannot pick the additional vector just anyway you wish. Use Sage effectively to help you.) By four such exchanges, you will complete the conversion.

For full marks, you should list all five bases (start, finish, and three intermediate) and clearly indicate which vector you removed and which you added. Your submitted work should show a computation at each stage that verifies that your four vectors really are a basis of the null space. The Sage `span()` constructor could be useful for doing this,