AIMS DMG Exercises 07
AIMS 2013-14: Designs, Matroids and Graphs

Rob Beezer
University of Puget Sound

Nancy Neudauer
Pacific University

January 7, 2014

Exercise 1. Build several interesting designs in Sage. Try `designs.(tab)` to see some possibilities. For each design, do the following:

1. The method `.parameters(t=2)` will report the values of t, v, k, λ when the design is considered as a 2-design.

2. Compute the incidence matrix. If you named your design `D`, then try `D.(tab)` to help locate a method that might do this for you. Read the help about viewing large matrices (more than 20 rows or 20 columns).

3. If you named your incidence matrix `A`, then compute the product of the incidence matrix with its transpose, AA^t. Again, `A.(tab)` might help you see how to do this.

4. Compute the parameters of the design again, only now consider the design as a 1-design, so pass in `t=1`.

You should now see a pattern to the matrix AA^t. (Much of mathematics is about recognizing patterns.) Explain what you see with a carefully-written paragraph, possibly using symbols to explain yourself. Go beyond just commenting on certain numbers showing up in certain places. How are the parameters of a design different when considered as a 2-design versus a 1-design? Using terms that describe the design, just what type of computation are we doing when we form AA^t? A complete answer will show that you understand the answers to both of these questions. Remember, a `shift-click` on a transient blue bar will bring up a LaTeX-enabled word processor. Use it.

To receive full marks submit evidence in your worksheet that you have considered several designs (steps 1-4 above) before formulating your ideas. Your paragraph about your observations needs to be clearly written, so that another student reading it would understand what you have discovered, and it should be clear that you can answer the two questions above.

Exercise 2. The cube graph, Q_3 is the skeleton of a cube. The next cell will create the graph, relabel its vertices with integers 0 through 7, and draw a resonably nice plot.

```
Q = graphs.CubeGraph(3)
Q.relabel()
Q.plot(layout='spring')
```

Now we provide two spanning trees of Q_3, T_0 and T_n (we do not know what n is yet). We check that they are really trees, are spanning trees, and are subgraphs. Read the help on `.is_subgraph()` to understand the order of the graphs and the use of the `induced` keyword.

```
T0 = Graph([(0,1),(1,5),(0,2),(2,3),(3,7),(0,4),(4,6)])
TN = Graph([(0,2),(2,3),(1,3),(3,7),(6,7),(5,7),(4,5)])
```

```
T0.is_tree(), T0.num_verts() == Q.num_verts(), T0.is_subgraph(Q, induced=False)
```

```
TN.is_tree(), TN.num_verts() == Q.num_verts(), TN.is_subgraph(Q, induced=False)
```

Your goal in this problem is to "exchange" edges to convert the one spanning tree into the other. More specifically, add an edge to a spanning tree, creating a cycle, and then remove an edge of the cycle to return to a (different) spanning tree. So your answer will be a list of trees, $T_0, T_1, T_2, \ldots, T_{n-1}, T_n$, so that T_i can be obtained from T_{i-1} by an exchange of edges.

The symmetric difference of the edge sets of T_0 and T_n might help you predict what n is. It might also be helpful in deciding exactly which edges to exchange.

```
E0 = Set(T0.edges(labels=False))
EN = Set(TN.edges(labels=False))
E0.symmetric_difference(EN)
```

Construct your trees as Sage graphs. Use Sage to verify that they are spanning trees. Verify that each adjacent pair in the list differs by an edge exchange by verifying that the symmetric difference of the edge sets of the two trees has cardinality 2. (Include T_0 and T_n in this check.) Python list comprehensions are a compact way to execute and present these checks.

Full marks will be given for a correct sequence of trees, and verification of their correctness with Sage commands.

Extra credit: what is the maximum number of edge exchanges necessary to convert between two spanning trees of Q_3? In other words, how big could n be (if the trees were different)? Give an argument for your answer.