C++ lab

Summer School: Computational aspects of buildings, UNCG

June 2019

Recall that the projective plane $A_2(k)$ is a graph whose vertices are the subspaces of k^3 of dimension 1 or 2 and we put an edge between two subspaces if one is included in the other. In the projective space $\mathbf{P}(k^3)$, 1-dimensional subspaces of k^3 become *points* and 2-dimensional subspaces *lines*. So we may think of a projective plane as a set of points P contained in various lines L satisfying the following properties:

- Any two distinct lines intersect in a unique point.
- Through any two distinct points passes a unique line.
- There are four points such that no lines contains three of them.

Eric Moorhouse has the list of known finite projective planes and files describing them `http://ericmoorhouse.org/pub/planes/`

1. **Problem:** Write a C++ program outputing a .txt file encoding the finite projective plane $A_2(\mathbf{F}_2)$.

There are 7 points and 7 lines in $A_2(\mathbf{F}_2)$, so we will use $\mathbf{Z}/7\mathbf{Z} = \{0, \ldots, 6\}$ as a model for both P and L. The adjacency is then given by the following lines:

$$L_0 = \{1, 2, 4\} \subset P = \mathbf{Z}/7\mathbf{Z},$$

$$L_i = L_0 + i = \{i + 1, i + 2, i + 4 \mod 7\}, \quad \text{with } i = 0, 1, \ldots, 6.$$

The output file should look like the file for $PG_2(2) = A_2(\mathbf{F}_2)$ available on Moorhouse's web page, (except he used $L_0 = \{0, 1, 3\}$.)

2. **Problem:** Same question for $A_2(\mathbf{F}_3)$.

The model here is $\mathbf{Z}/13\mathbf{Z} = \{0, \ldots, 12\}$, with $L_0 = \{1, 2, 5, 7\}$.

3. **Problem:** In the both previous examples, use the point-line correspondence $\lambda_0 : x \mapsto x$ and encode the graph G_λ with vertex set P where we put an oriented edge between x and y is $x \in \lambda(y)$.