CHMC Advanced:
Chip Firing

December 7, 2019

A **graph** G is a set of vertices $\{1, 2, ..., n\}$, together with a set of edges connecting pairs of vertices. If two vertices i, j have an edge between them, we call the edges **adjacent**.

Chip firing consists of the following game:

1. At the start of the game, every vertex has a set (non-negative) number of "chips". We will write c_i for the number of chips at vertex i.

2. Pick a vertex i with at least as many chips on i as there are neighbors of i.

3. Transfer 1 chip from i to each of i's neighbors, so that c_i becomes $c_i - \#\text{neighbors}$, and for each neighbor j, c_j becomes $c_j + 1$. This process is called **firing** vertex i.

4. Go back to step 2.

Some problems:

1. Construct an example of a chip firing game which eventually cannot be fired. (Specify how many vertices, how many edges, and how many total chips your graph has.)

2. Construct an example of a chip firing game which can be played infinitely often, i.e. you can always find a chip that fires.

3. If a chip firing game is infinite, then every vertex is fired infinitely often; why?

4. If a chip firing game is finite, then there is a vertex that is never fired; why?

5. Suppose that G has n vertices, m edges, and N total chips. Show that:

   - if $N > 2m - n$, then the game is infinite.
   - if $m \leq N \leq 2m - n$, then there is an initial chip configuration such that the game is finite, as well as an initial chip configuration such that the game is infinite.
   - (Harder) if $N < m$, then the game is finite.

For the last few problems, see what happens with actual chip firing games, such as on a complete graph, cyclic graph, or tree.