1 Sage Can . . .

1. Take any Coxeter matrix and create the Coxeter group, and this can be done in the coxeter3 implementation; infinite weights can be input as -1 in the matrix.
2. Specify generator symbols hen you create a Coxeter group, or change the symbols after the group created.
3. Create an element using a word (the command is called "from reduced word", but the input doesn't have to be reduced). Then you can get all the reduced words of that element as well as the reduced word graph. (Not sure if this works for all groups (AG) or just a small class of nice groups.)
4. Generate all elements of a given length in a Coxeter group. (AG?)
5. Compute the left and right descent sets of any $w \in W$. (AG?)
6. Check if an affine permutation is FC; involves lehmer codes.

2 Sage Cannot . . .

3 Todo

1. Write a function to tell if a given word is reduced, from a Coxeter matrix.
2. Write a function to check if a (reduced) word is fully commutative. To do this we may need to earn about lehmer codes and how FC is checked for affine permutations. We may also need to learn how reduced word graphs are produced, since an element is FC iff there's only one commutation-connected component in the reduced word graph.
3. Understand how Sage picks a canonical reduced word for each element.