Question

A typical bicycle drivetrain has a number of cogs on front crank (where the rider pedals) and a number of cogs on the rear cassette (on the wheel). The output ratio from the pedals to the rear wheel is:

tooth count on selected front cog / tooth count on selected rear cog

All cogs have an integer number of teeth and only a single cog on the front and single cog on the rear can be selected at one time. Preferably code this in Python, but other languages would be acceptable.

For a given set of front and rear cogs:

1. write a program that determines the gear combination providing the closest ratio that is less than or equal to the target ratio.

The function might look something like this: get\_gear\_combination(f\_cogs, r\_cogs, target\_ratio)

For an example input of:

f\_cogs = [38, 30]

r\_cogs = [28, 23, 19, 16]

target\_ratio = 1.6

get\_gear\_combination(f\_cogs, r\_cogs, target\_ratio)

should print:

Front: 30, Rear: 19, Ratio 1.579

1. write a program that determines a shift sequence to traverse from an initial gear combination to a gear combination with the closest ratio. Each shift can only change one gear on either the front or rear.

The function might look something like this: get\_shift\_sequence(f\_cogs, r\_cogs, ratio, initial\_combination)

For example, input  
f\_cogs = [38, 30]  
r\_cogs = [28, 23, 19, 16]  
ratio = 1.6  
initial\_combination = (F:38 R:28)

get\_shift\_sequence(f\_cogs, r\_cogs, ratio, initial\_combination)

should print:

1 - F:38 R:28 Ratio 1.357

2 - F:30 R:28 Ratio 1.071

3 - F:30 R:23 Ratio 1.304

4 - F:30 R:19 Ratio 1.579