Brian Lins discusses the algorithm for defining a fair car pool system in his article for Math Horizons, November 2015, “Whose Turn It It to Drive Today?” The naïve system that his carpool had been using was clunky, and could easily lead to confusion and feelings of unfair contributions.

By using a greedy algorithm, participants are awarded or deducted “points” for their roles in the carpool. For each day, or iteration of the algorithm, the driver is decided based on who has the fewest points banked. The points added to the driver and deducted from a passenger have a net balance of zero. Given p is the number of people participating that day, then we can say that a passenger will pay the driver 1/p of their points and a driver will receive (p-1)/p points. To avoid fractions, the writer uses the least common multiple (LCM) to give everyone an integer point value.

The trick to this algorithm is to discount those people who are not participating on any given day. For this, we start by asking if each person is participating and getting a ‘p’ value for our calculations. We also need to know the LCM, but that is easy enough to calculate. Assuming a car maxes out at 5 passengers and a van at 6, then the LCM holds at 60.

Pseudocode:

Vector<person> carpool;

Int p =0;

For each person in carpool

Is person participating?

If yes

Set participate = True

p++

For each person in carpool

If participate == True

If pts are min

Display person is driver

Pts += LCM\*(p-1)/p

Else

Pts -= LCM\*(1/p)

This process is greedy because at each point, the algorithm choses the one person participating for the day with the lowest score to be the driver in hopes of striking a balance and minimizing the difference between in points between all persons in the carpool.