Source: ⟦KBe2020math401refExr0nRetIndex⟧

I think a good math problem is one that takes you somewhere and reveals something useful and elegant. Jana talks about math as a game, where we define the set of rules based on what makes the game fun or useful. I love this analogy because it reminds me of a particular set of rules that are dear to me, the rules surrounding computational complexity analysis.

In that universe, "constant factors" aka coefficients don't matter, and we simply drop them when presenting an answer. Although this may feel wrong at first glance, it makes lots of sense in the one place where it's used–because we needed a way to evaluate algorithms based on not exact speed but how complex they were, since the exact amount of time or number of steps required might change as hardware improves or becomes abstract. My favorite example of the beauty of this has to do with a data structure called disjoint set union. It is a lazy propagation structure that stores temporary data to remember what sets have been merged. The proof for the complexity of this algorithm takes advantage of figuring out how much we expect the structure to grow, and end up with it being roughly constant time. I would also make a note about how Djikstra's time complexity somewhat hijacks the system by taking advantage of log's power to coefficient manipulations, but this is getting kind of long. My point is, a good math problem/project is a set of rules that do something interesting, elegant, and useful, especially if you might not have expected it.

As you may have gathered from the previous paragraph, I am excited about computers and algorithms. I also enjoy mountain biking and fiddling with my workflow.