Dylan Mohsen
DS210 Hw 6

1. Using cargo run, the computation time increases significantly as k grows. For k=46, it took about 46.245 seconds to compute the Fibonacci number. The growth in computation time is exponential, reflecting the exponential increase in function calls due to the use of recursion.
   cargo run --release mode shows an improvement in computation times. For k=50, it took about 39.911616 seconds to compute the Fibonacci number. The optimizations make the computation much more efficient, but still has the same exponential growth trend.
   Comparing the times for k=46:
   Cargo run: ~46.245 seconds
   Cargo run –release: ~5.850782 seconds

   The multiplicative difference for k=46 is roughly 46.245 / 5.850782 ≈ 7.9, indicating that the computation in release mode is nearly 8 times faster for this value of K.

2. When trying to run the modified code with u8 using cargo run results in an error of an attempt to add with overflow and stops the function and panics.

   However, when using cargo run –release, it inputs values but the results are restricted to values from numbers 0 to 255 as this is all u8 can represent as values.

3. The reason I chose to use u64 is using u32 would not be able to properly store all possible values of u32 squared closer to its limit, and u64 is able to hold much larger values than the largest u32 number squared.