Azalea Rohr

Homework 6

March 18, 2023

Collaboration: None

Question 1

- With cargo run, as the values of k got bigger the run time got exponentially longer and had to run for multiple minutes in order to run until computation. When cargo run --release was used to run the program, only a few seconds were needed to run the whole program, but as k got bigger the run time still got larger as well. So using cargo run --release optimizes the run time drastically including for the larger values of k.
- Below are the values for using cargo run (up to k=48)
    - k: 0, Fk: 0, Time: 0ns
    - k: 1, Fk: 1, Time: 0ns
    - k: 2, Fk: 1, Time: 0ns
    - k: 3, Fk: 2, Time: 0ns
    - k: 4, Fk: 3, Time: 0ns
    - k: 5, Fk: 5, Time: 0ns
    - k: 6, Fk: 8, Time: 0ns
    - k: 7, Fk: 13, Time: 0ns
    - k: 8, Fk: 21, Time: 1µs
    - k: 9, Fk: 34, Time: 1µs
    - k: 10, Fk: 55, Time: 1µs
    - k: 11, Fk: 89, Time: 3µs
    - k: 12, Fk: 144, Time: 4µs
    - k: 13, Fk: 233, Time: 8µs
    - k: 14, Fk: 377, Time: 13µs
    - k: 15, Fk: 610, Time: 16µs
    - k: 16, Fk: 987, Time: 26µs
    - k: 17, Fk: 1597, Time: 40µs
    - k: 18, Fk: 2584, Time: 73µs
    - k: 19, Fk: 4181, Time: 120µs
    - k: 20, Fk: 6765, Time: 190µs
    - k: 21, Fk: 10946, Time: 283µs

  - k: 22, Fk: 17711, Time: 460µs
  - k: 23, Fk: 28657, Time: 763µs
  - k: 24, Fk: 46368, Time: 1.64ms
  - k: 25, Fk: 75025, Time: 3.833ms
  - k: 26, Fk: 121393, Time: 3.226ms
  - k: 27, Fk: 196418, Time: 5.589ms
  - k: 28, Fk: 317811, Time: 10.065ms
  - k: 29, Fk: 514229, Time: 14.803ms
  - k: 30, Fk: 832040, Time: 24.094ms
  - k: 31, Fk: 1346269, Time: 33.253ms
  - k: 32, Fk: 2178309, Time: 50.973ms
  - k: 33, Fk: 3524578, Time: 82.738ms
  - k: 34, Fk: 5702887, Time: 132.995ms
  - k: 35, Fk: 9227465, Time: 214.833ms
  - k: 36, Fk: 14930352, Time: 347.597ms
  - k: 37, Fk: 24157817, Time: 563.934ms
  - k: 38, Fk: 39088169, Time: 913.787ms
  - k: 39, Fk: 63245986, Time: 1.495054s
  - k: 40, Fk: 102334155, Time: 2.40098s
  - k: 41, Fk: 165580141, Time: 3.895769s
  - k: 42, Fk: 267914296, Time: 6.270682s
  - k: 43, Fk: 433494437, Time: 10.169893s
  - k: 44, Fk: 701408733, Time: 16.393887s
  - k: 45, Fk: 1134903170, Time: 26.782078s
  - k: 46, Fk: 1836311903, Time: 43.052539s
  - k: 47, Fk: 2971215073, Time: 69.440269s
  - k: 48, Fk: 4807526976, Time: 112.54297s
- Below are the values using cargo run --release (up to k=48)
  - k: 0, Fk: 0, Time: 0ns
  - k: 1, Fk: 1, Time: 0ns
  - k: 2, Fk: 1, Time: 0ns
  - k: 3, Fk: 2, Time: 0ns
  - k: 4, Fk: 3, Time: 0ns
  - k: 5, Fk: 5, Time: 0ns

- k: 6, Fk: 8, Time: 0ns
- k: 7, Fk: 13, Time: 0ns
- k: 8, Fk: 21, Time: 1µs
- k: 9, Fk: 34, Time: 1µs
- k: 10, Fk: 55, Time: 0ns
- k: 11, Fk: 89, Time: 1µs
- k: 12, Fk: 144, Time: 1µs
- k: 13, Fk: 233, Time: 2µs
- k: 14, Fk: 377, Time: 3µs
- k: 15, Fk: 610, Time: 4µs
- k: 16, Fk: 987, Time: 6µs
- k: 17, Fk: 1597, Time: 10µs
- k: 18, Fk: 2584, Time: 17µs
- k: 19, Fk: 4181, Time: 28µs
- k: 20, Fk: 6765, Time: 45µs
- k: 21, Fk: 10946, Time: 71µs
- k: 22, Fk: 17711, Time: 126µs
- k: 23, Fk: 28657, Time: 184µs
- k: 24, Fk: 46368, Time: 312µs
- k: 25, Fk: 75025, Time: 485µs
- k: 26, Fk: 121393, Time: 784µs
- k: 27, Fk: 196418, Time: 1.503ms
- k: 28, Fk: 317811, Time: 2.304ms
- k: 29, Fk: 514229, Time: 3.177ms
- k: 30, Fk: 832040, Time: 7.96ms
- k: 31, Fk: 1346269, Time: 9.854ms
- k: 32, Fk: 2178309, Time: 13.018ms
- k: 33, Fk: 3524578, Time: 17.05ms
- k: 34, Fk: 5702887, Time: 22.872ms
- k: 35, Fk: 9227465, Time: 31.644ms
- k: 36, Fk: 14930352, Time: 49.059ms
- k: 37, Fk: 24157817, Time: 78.865ms
- k: 38, Fk: 39088169, Time: 127.709ms
- k: 39, Fk: 63245986, Time: 205.764ms

- k: 40, Fk: 102334155, Time: 332.7ms
- k: 41, Fk: 165580141, Time: 539.364ms
- k: 42, Fk: 267914296, Time: 937.856ms
- k: 43, Fk: 433494437, Time: 1.439341s
- k: 44, Fk: 701408733, Time: 2.318722s
- k: 45, Fk: 1134903170, Time: 3.945409s
- k: 46, Fk: 1836311903, Time: 6.028371s
- k: 47, Fk: 2971215073, Time: 9.963604s
- k: 48, Fk: 4807526976, Time: 16.230282s

Question 2

- With cargo run --release my run time was faster, however, both of these programs did work for me even with u8. With the u128 numbers, they went up to 354224848179261915075 with the largest number, however with u8 the largest number was 245 since the largest number that a u8 can hold is 2^8 (256). So with the u8 numbers, the program was not able to fully run to completion.

Question 3

- Within this code, a non-negative integer K is read from the user's input, then a 'for' loop is used to iterate from i =1 to k and computes the sum of squares of i. To make sure there is not integer overflow, I used the 'u128' type to make sure the range is large enough to hold the sum of squares for the given range of k. I cast i to u128 before doing the multiplication to make sure that the results do not go out of the range of u128. Integer overflow is not a problem within this code since u128 provides a large range of values from 0 to $2^{128} - 1$, so I can compute the sum of squares for large values of k. Since any input will be a u32 which has a smaller capacity than a u128, meaning that the u128 will be able to hold a u32 easily.