

Hrithik Shivrath
DS210
Prof Leonidas Kontothanassis

Report Q1: What are the computation times for different k for each of the options? How do the times to compute Fibonacci numbers compare for large k between these two options? Are they roughly the same or are they very different? If they are different, what is the multiplicative difference?

The computation times for run and run release are different. This is because run would check if individual operations and variables are compatible before completing the running of the program while run release would run the program without that check for each of the operations and variations in the program. As we can see there is about a 8 times multiplicative difference between cargo run and cargo run release. It run

```
Time it took: 934.922ms  
the fibonacci sequence for 39 is 63245986  
Time it took: 1.512566s  
the fibonacci sequence for 40 is 102334155  
Time it took: 2.444068s  
the fibonacci sequence for 41 is 165580141  
Time it took: 3.960933s  
the fibonacci sequence for 42 is 267914296  
Time it took: 6.447782s  
the fibonacci sequence for 43 is 433494437  
Time it took: 10.500631s  
the fibonacci sequence for 44 is 701408733  
Time it took: 16.832952s  
the fibonacci sequence for 45 is 1134903170  
Time it took: 27.284917s  
the fibonacci sequence for 46 is 1836311903  
Time it took: 44.494077s
```

```
the fibonacci sequence for 41 is 165580141  
Time it took: 549.387ms  
the fibonacci sequence for 42 is 267914296  
Time it took: 887.316ms  
the fibonacci sequence for 43 is 433494437  
Time it took: 1.44172s  
the fibonacci sequence for 44 is 701408733  
Time it took: 2.349001s  
the fibonacci sequence for 45 is 1134903170  
Time it took: 3.78143s  
the fibonacci sequence for 46 is 1836311903  
Time it took: 6.155453s  
the fibonacci sequence for 47 is 2971215073  
Time it took: 10.500631s
```

Report: Now conduct the following experiment. Replace the array entry type with u8 and adjust any other types accordingly so your program still compiles. Try running the modified code with both cargo run and cargo run --release. Are there any differences in the behavior of the program? If so, what are they?

We can notice that when the u8 is used instead of the u128 the array of fibonacci numbers is not able to fit as many bytes since the u8 takes a smaller number of bytes so as you notice the larger number which appear at the end of the u128 array gets cut off to smaller numbers in the u8 array because of this overflow issue.

```
2 | let mut F:[u128; 101] = [0;101];
  | ^ help: convert the identifier to snake case (notice the capitalization): `f`

warning: `Q2` (bin "Q2") generated 2 warnings
  Finished dev [unoptimized + debuginfo] target(s) in 0.37s
  Running `target/debug/Q2`
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025, 20365011074, 32951280099, 53316291173, 86267571272, 139583862445, 225851433717, 365435296162, 591286729879, 956722026041, 1548008755920, 2504730781961, 4052739537881, 6557470319842, 10610209857723, 17167680177565, 27777890035288, 44945570212853, 72723460248141, 117669030460994, 190392490709135, 308061521170129, 498454011879264, 806515533049393, 1304969544928657, 2111485077978050, 3416454622906707, 5527939700884757, 8944394323791464, 14472334024676221, 23416728348467685, 37889062373143906, 61305790721611591, 99194853094755497, 160500643816367088, 259695496911122585, 420196140727489673, 679891637638612258, 110008778366101931, 1779979416004714189, 2880067194370816120, 4660046610375530309, 7540113804746346429, 12200160415121876738, 19749274219868223167, 31940434634990099905, 51680708854858323072, 83621143489848422977, 135301852344706746049, 218922995834555169026, 354224848179261915075]
(base) gshiv@crc-dot1x-nat-10-239-196-143 Q2 %
```

```
Compiling Q2 v0.1.0 (/Users/gshiv/projects/HW6/Q2)
warning: crate `Q2` should have a snake case name
  |
  | = help: convert the identifier to snake case: `q2`
  | = note: `#[warn(non_snake_case)]` on by default
warning: variable `F` should have a snake case name
--> src/main.rs:2:13
2 | let mut F:[u8; 101] = [0;101];
  | ^ help: convert the identifier to snake case (notice the capitalization): `f`

warning: `Q2` (bin "Q2") generated 2 warnings
  Finished dev [unoptimized + debuginfo] target(s) in 0.18s
  Running `target/debug/Q2`
thread 'main' panicked at src/main.rs:5:14:
attempt to add with overflow
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
(base) gshiv@crc-dot1x-nat-10-239-196-143 Q2 %
```

```
= help: convert the identifier to snake case: `q2`
= note: `#[warn(non_snake_case)]` on by default
warning: variable `F` should have a snake case name
--> src/main.rs:2:13
2 | let mut F:[u8; 101] = [0;101];
  | ^ help: convert the identifier to snake case (notice the capitalization): `f`

warning: `Q2` (bin "Q2") generated 2 warnings
  Finished release [optimized] target(s) in 0.12s
  Running `target/release/Q2`
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 121, 98, 219, 61, 24, 85, 109, 194, 47, 241, 32, 17, 49, 66, 115, 181, 40, 221, 5, 226, 231, 201, 176, 121, 41, 162, 203, 109, 56, 165, 221, 130, 95, 225, 64, 33, 97, 130, 227, 101, 72, 173, 245, 162, 151, 57, 208, 9, 217, 226, 187, 157, 88, 245, 77, 66, 143, 209, 96, 49, 145, 194, 83, 21, 104, 125, 229, 98, 71, 169, 240, 153, 137, 34, 171, 205, 120, 69, 189, 2, 191, 193, 128, 65, 193, 2, 195]
(base) gshiv@crc-dot1x-nat-10-239-196-143 Q2 %
```

Report: Explain why the situation described above is not happening, i.e., why the range of integers you use is sufficiently large. This kind of problem is known as integer overflow, i.e., you want to explain why integer overflow is not a problem in your code.

If you input the highest u32 value, it would $2^{32}-1$, the output for this function would be less than the highest u128, which is $2^{128}-1$, therefore it can be safely represented as a u128.

```
For more information about this error, try `rustc --explain E0433`.
error: could not compile `Q3` (bin "Q3") due to 1 previous error
(base) gshiv@crc-dot1x-nat-10-239-196-143 Q3 % cargo run
  Compiling Q3 v0.1.0 (/Users/gshiv/projects/HW6/Q3)
warning: crate `Q3` should have a snake case name
|
= help: convert the identifier to snake case: `q3`
= note: `[warn(non_snake_case)]` on by default

warning: `Q3` (bin "Q3") generated 1 warning
  Finished dev [unoptimized + debuginfo] target(s) in 0.38s
  Running `target/debug/Q3`
```

3

14