Pseudocode

```
PROMPT for fib_num
GET fib_num

fib1 <- 1
fib2 <- 0
fib3 <- 1

FOR i <- 1 … fib_num   # A
     fib1 += fib2      # B
     fib2 <- fib3      # C
     fib3 <- fib1      # D

PUT fib1
```

Program Trace

	fib_num	i	fib1	fib2	fib3
A	7	1	1	0	1
B	7	1	1	0	1
C	7	1	1	1	1
D	7	1	1	1	1
A	7	2	1	1	1
B	7	2	2	1	1
C	7	2	2	1	1
D	7	2	2	1	2
A	7	3	2	1	2
B	7	3	3	1	2
C	7	3	3	2	2
D	7	3	3	2	3
A	7	4	3	2	3
B	7	4	5	2	3
C	7	4	5	3	3
D	7	4	5	3	5
A	7	5	5	3	5
B	7	5	8	3	5
C	7	5	8	5	5
D	7	5	8	5	8
A	7	6	8	5	8
B	7	6	13	5	8
C	7	6	13	8	8
D	7	6	13	8	13
A	7	7	13	8	13

Algorithmic Efficiency

This algorithm is *O(n)*. The reason being is that there is only one FOR Loop. This FOR loop runs dependent on the input of the user, but we visit every element in the collection once. Doubling the input simply means we double the amount of time units.