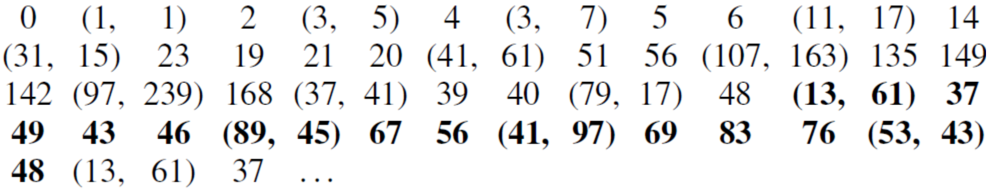
CS 7A - Spring 2015 - Richard Guy's Sub-Fibonacci Sequence. Due 5/21/15

Subprime Fibonacci sequences

0.1 Background Theory

Start like the Fibonacci sequence $0, 1, 1, 2, 3, 5, \ldots$, but before you write down a composite term, divide it by its least prime factor so that this next term is not 8, but rather $8/2 = 4$. After that the sum gives us $5 + 4 = 9$, but we write $9/3 = 3$; then $4 + 3 = 7$, which is okay since it is prime; then $3 + 7 = 10$, but we write $10/2 = 5$; and so on:



0 (1, 1) 2 (3, 5) 4 (3, 7) 5 6 (11, 17) 14
(31, 15) 23 19 21 20 (41, 61) 51 56 (107, 163) 135 149
142 (97, 239) 168 (37, 41) 39 40 (79, 17) 48 **(13, 61) 37**
49 43 46 (89, 45) 67 56 (41, 97) 69 83 76 (53, 43)
48 (13, 61) 37 ...

and we are in an 18-cycle. If we start with $1, 1$ or $1, 2$, it follows that we get the same result. But we may start with any pair of numbers, and you may like to try starting with $2, 1$, or $1, 3$, or $3, 9$, or $13, 11$, etc.

One might suspect that every such sequence enters this 18-cycle, similar to the Collatz conjecture. After all, since our sequences are bounded or unbounded, they must either enter a cycle or increase indefinitely. We do not believe the latter happens, and we provide a heuristic argument below. But is the 18-cycle the only "non-trivial" cycle? Wait and see.

First, note that a, a, where a $a \neq \pm 1$, gives the sequence $a, a, a, a, \ldots$, This is a trivial cycle. Sequences that end in trivial cycles are trivial sequences, such as $5, 15, 10, 5, 5, 5, \ldots$, or $-143, 39, -52, -13, -13, -13, \ldots$, If two consecutive terms have the same sign, then so do all subsequent terms. If they have opposite sign or include a zero, they bound further terms until two consecutive terms of the same sign appear, e.g., $-17, 7, -5, 2, -3, -1, -2, \ldots$, after which the sign remains constant.

Next, two terms of opposite parity are followed by an odd term, and two odd terms are followed by an even or an odd term depending on whether their sum is a multiple of 4. One can have arbitrarily long strings of even terms, but they must terminate since the power of 2 in consecutive terms must eventually decrease, e.g., $128, 160, 144, 152, 148, 150, 149, \ldots,$; once we have an odd term (unless this sequence is trivial), subsequent even terms are isolated with each followed by at least two odd terms. Therefore, *we need only consider sequences of positive terms, comprised of "runs" of odd terms separated by even terms.*

Finally, let the shape of a sequence be the string of its terms' parities (O for odd, E for even). The Fibonacci sequence has shape EOOEOOEOOEOO... Our first subprime Fibonacci sequence had shape EOOEOOEOOOEOO... The "extra" odd term here came from where the sum of the previous two odd terms had only one factor of 2. The example, starting at $13, 61$ inclusive, gives the shape OOOOOEOOOEOOOOEOOE that repeats with the 18-cycle.

0.2 Nodes and other cycles

Our sequences cannot immediately be represented in the fashion of the Collatz Conjecture because of the second-order nature of our recurrence. We must carefully define vertices for our sequences, and so we introduce two important terms. The *nodes* of a sequence are ordered pairs of positive, odd coprimes that either begin the sequence or immediately follow the even terms of a sequence. *Runs* are the strings that begin with a node and consist of odd terms together with a single terminating even term.
In the next section we will see that every non-trivial sequence becomes composed of runs after some point. Here

is our initial sequence with nodes parenthesized:

(151, 227) 189 208 (397, 121) 259 190 (449, 213) 331 272 (201, 43) 122
(55, 59) 57 58 **(23, 27) 25 26 (17, 43) 30 (73, 103) 88**
(191, 93) 142 (47, 63) 55 59 57 58 (23, 27) …

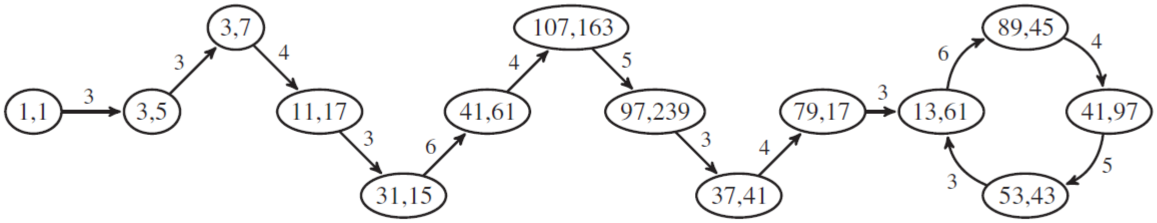


Figure 2 Path generated by the 0, 1 sequence

Taking the sequence's shape, we can treat each substring of the form O...OE as a unit, starting when the first two terms of such a substring are coprime and not preceded by an odd term. The corresponding terms comprise a run, and the first two terms of the run comprise a node. Let us now construct our first sequence path (FIGURE 2 above). For notational convenience we weight the digraph by assigning to each arc the length of the run generated by the node at the arc's tail. We could then imagine the infinite digraph generated by all non-trivial subprime Fibonacci sequences, as is done for the Collatz sequences. If the 18-cycle were the only non-trivial cycle, the subprime Fibonacci digraph would look like FIGURE 3.

One reason this digraph is a nice representation is that it shows how many nodes are *direct predecessors* to a single node. If a node is a predecessor (not necessarily direct) to a node or cycle, we say it is *tributary* to the node or cycle. How could we grow this graph? One way is to go outwards from known nodes. This would require a way of enumerating a node's direct predecessors, which can be done with some work:

1. For example, with the node $(89, 45)$ of the 18-cycle, start with the preceding even term t, which must satisfy $t + 89 = 45q$, where q is 1 or 3 ($q = 2$ makes t odd, and q cannot exceed a prime factor of 45). This gives $t = -44$ or 46, so the node must always be preceded by 46.

2. Let the positive odd term before t, if it exists, be s. Then $s + t = 89p$, where p is 1 or an odd prime ≤ 89. For $t = 46$, possible values of s are $43, 221, 399$, etc. The term before s must also be odd. If this term is r, it must satisfy $r + s = 2t$ since $r + s$ is even. For example, $s = 43$ gives $r = 49$, and none of the other possibilities for s would work, since they would make $r \leq 0$.

3. Since we are only looking for possible direct predecessors (positive, odd coprimes), we can assume that each prior step involved division by two. Working backwards gives $\ldots, -83, 109, 13, 61, 37, 49, 43, 46$. Thus, our direct predecessors are exactly $(109, 13), (13, 61), (61, 37), (37, 49)$, and $(49, 43)$, only two of which are depicted in FIGURE 3 below.

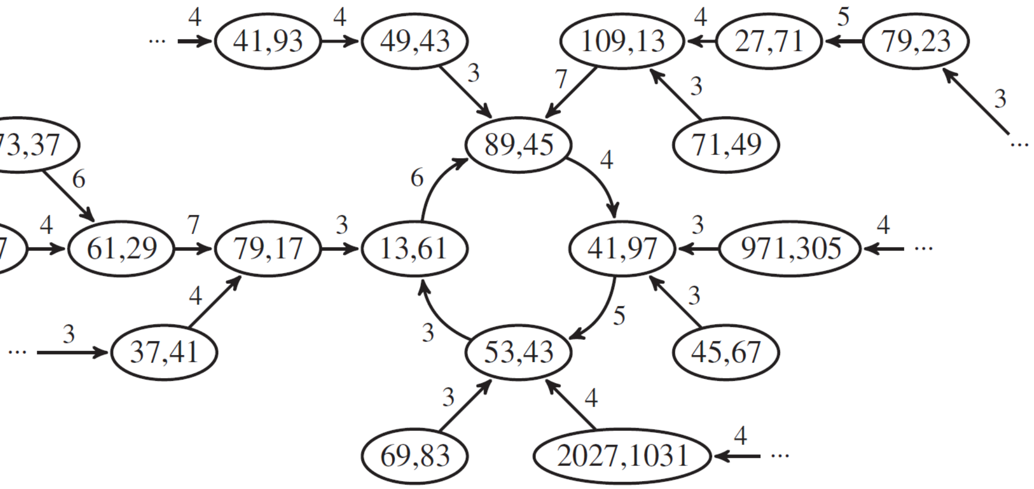
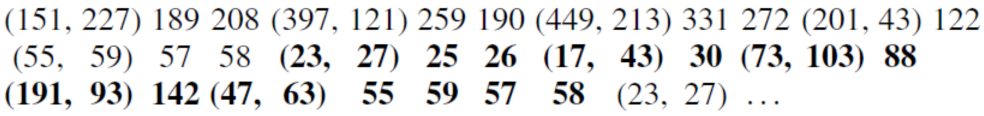


Figure 3 Some paths leading to the 18-cycle

Contrast this with working the Collatz conjecture, where there are at most two direct predecessors as a result of the sequence definition. This procedure for constructively generating nodes is quite finicky, however, and discourages a graph-theoretic approach to analysis. Not to say that it is impossible; there exist reductions and results on the Collatz graph, and we encourage the reader to explore the possibility of deriving properties for the subprime digraph from this perspective.

Do sequences all enter the 18-cycle we have already seen? In other words, is the subprime digraph weakly connected? Let us start at the node $(151, 227)$:


(151, 227) 189 208 (397, 121) 259 190 (449, 213) 331 272 (201, 43) 122
(55, 59) 57 58 **(23, 27) 25 26 (17, 43) 30 (73, 103) 88**
(191, 93) 142 (47, 63) 55 59 57 58 (23, 27) ...

and we are in a 19-cycle whose first repeated node is $(23, 27)$. Note that though $55, 59$ are the first two repeated terms, they only act as a node the first time through; thus $(47, 63)$ being a node with the terms $55, 59$ in its run does not preclude $(55, 59)$ from being a node in another context. Both nodes are tributary to the node $(23, 27)$.

Furthermore, if you start with $(5, 13)$ you will enter a 136-cycle through node $(47, 23)$ (though simpler starting terms like $1, 4$ suffice). If you start with $5, 23$ you will enter a 56-cycle through node $(119, 109)$ with 5693 as its largest term. Finally, the nodes $(37, 199)$ and $(127, 509)$ generate an 11-cycle and a 10-cycle, respectively.

FIGURE 4 displays the nodes in these non-trivial cycles. We checked sequences that start with two numbers below 106 and found no non-trivial cycles other than these six. That bound is easily extendable by our more computationally-minded readers.

In TABLE 1 the headings indicate the range for the first two terms of the sequence, and the entries are the number of occurrences for each cycle length. The proportion of pairs that generate each non-trivial cycle stabilizes

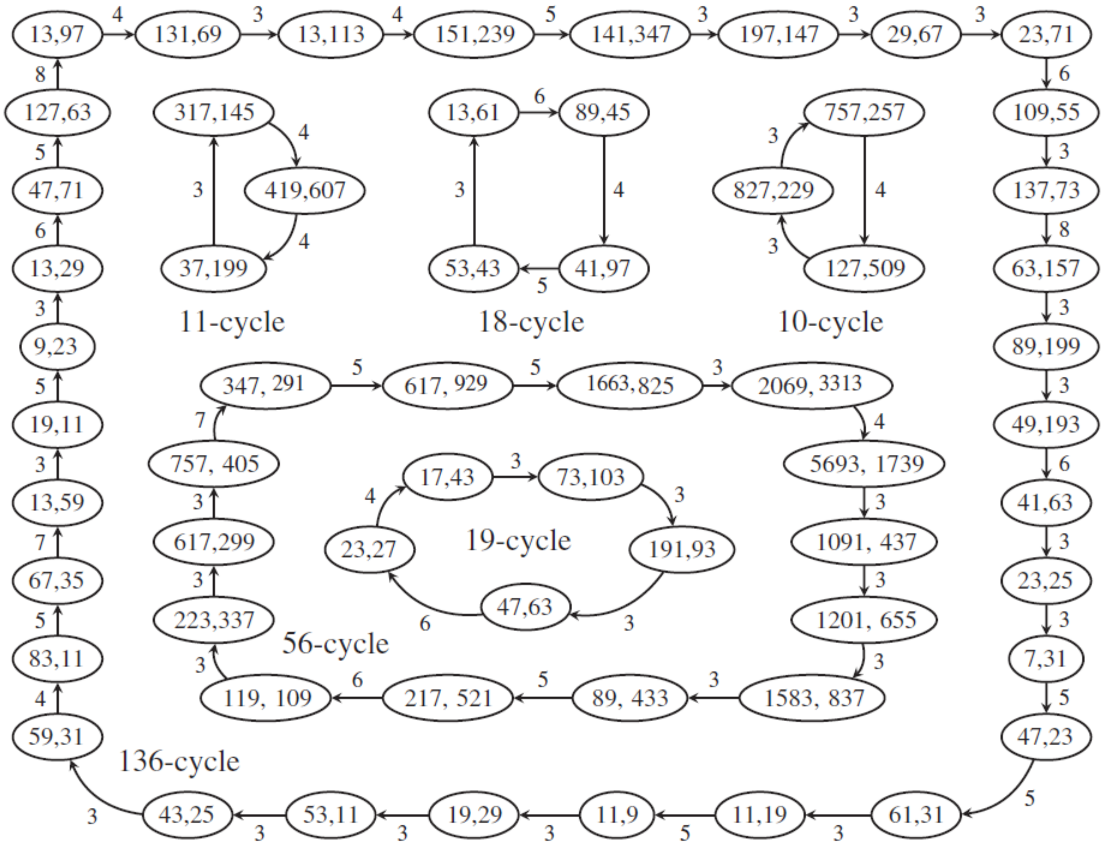


Figure 4 Digraphs of the six known non-trivial cycles

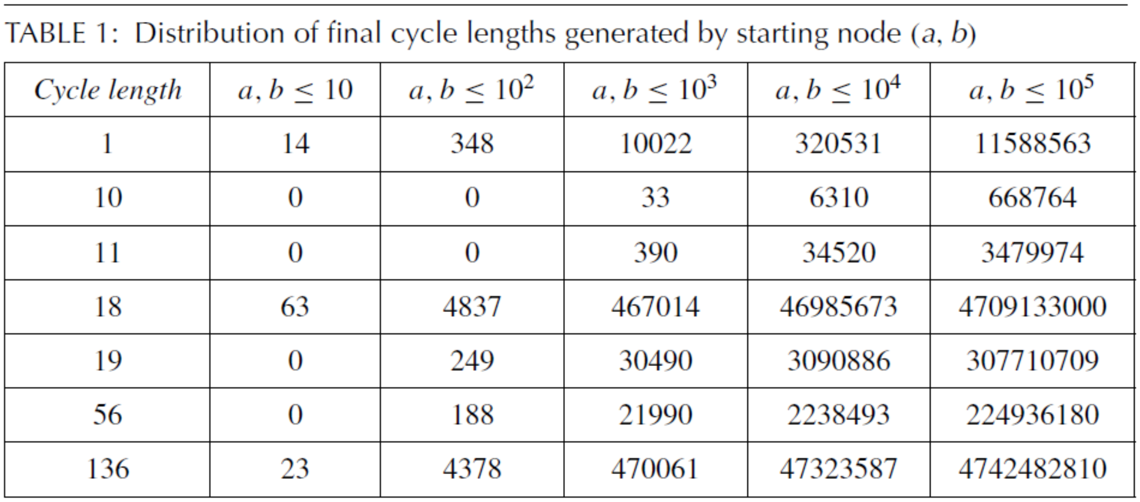


TABLE 1: Distribution of final cycle lengths generated by starting node (a, b)

Cycle length	$a, b \leq 10$	$a, b \leq 10^2$	$a, b \leq 10^3$	$a, b \leq 10^4$	$a, b \leq 10^5$
1	14	348	10022	320531	11588563
10	0	0	33	6310	668764
11	0	0	390	34520	3479974
18	63	4837	467014	46985673	4709133000
19	0	249	30490	3090886	307710709
56	0	188	21990	2238493	224936180
136	23	4378	470061	47323587	4742482810

as the range for starting terms increases. Additionally, non-trivial cycles appear to be distributed among the starting pairs rather arbitrarily. However, trivial cycles decrease in proportion since a cycle $a, a, \ldots$ requires all earlier terms to be multiples of a. Applying the "direct predecessor" method shows why this is, and how this makes relatively few starting conditions lead to a given trivial cycle.

Generally, non-trivial sequences seem to exhibit pseudo-random behavior in their terms and their digraphs, regarding the length of their paths, the nodes they pass, and their associated cycles. We believe this is due partly to the construction, which relies on prime factorizations (the relationship between these factorizations and addition is not well understood). A similar difficulty is seen in the earlier RATS sequences, where the relationship between base-dependent reversal/sort and addition is essential to analysis.

However, another source of apparent randomness seems to be the iteration's conditionality itself, as with the Collatz sequences. For example, if one considers a variant of subprime Fibonacci where only division by 5 occurs (when the sum is divisible by 5), similar observations as the above arise. We begin to feel the apparent intractability mentioned earlier of proving results on "destinies" of sequences like these.

1 Problems

We developed the `class` Fibonacci described below in lab:

```
///The  Fibonacci Sequence
#include "std_lib_facilities.h"

class Fibonacci {
public:
    unsigned long long seed1;
    unsigned long long seed2;
    unsigned fibSize;
    vector<unsigned long long> fibVec;
    Fibonacci(unsigned long long,
              unsigned long long,
              unsigned);
    void print();
};

Fibonacci::Fibonacci(unsigned long long a, unsigned long long b, unsigned sz)
    : seed1(a), seed2(b), fibSize(sz) {
    fibVec.push_back(seed1);
    fibVec.push_back(seed2);
```

```
    for(int i = 0; i < fibSize-2; ++i)
        fibVec.push_back(fibVec[i]+fibVec[i+1]);
}

void Fibonacci::print() {
    for( unsigned long long v : fibVec ) cout << v << '␣';
}
```

Here then is the driver function, `main()`:

```
///GH: Fibonacci

int main() {
    Fibonacci f(0,1,100);
    f.print();
}
```

Which produces this output:

```
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 57 02887 9227465 14930352 24157817 39088169 63245986 102334155
165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073
4807526976 7778742049 12586 269025 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 1100087778366101931
1779979416004714189 2880067194370816120 4660046610375530309
7540113804746346429 12200160415121876738 1293530146158671551
13493690561280548289 14787220707439219840 9834167195010216513
```

1. Modify the `class` for a class `SubPrimeFib` that includes data members

   ```
   unsigned long long seed1;  //input 1
   unsigned long long seed2;  //input 2
   unsigned fibSize;    // the length of a sequence including one cycle
   unsigned cycleLength;  // the length of the cycle
   vector<unsigned long long> fibVec;  // a sequence including one cycle
   vector<unsigned long long> fibCycle; // the cyclic part of the sequence
   vector<Noderun> subPrimeNodes; //sequence of nodes and runs in the cycle
   ```

   and member functions

   ```
   subPrimeFib(unsigned long long,
             unsigned long long);///Constructor does most of the work
   void print(); // print sequence with first cycle
   void printNoderuns(); // print node-run, (a,b)-r-, format for cycle
   void printCycle(); //print one cycle and define cycle vector
   ```

   and helper function

```
// check to see if a vector is looping yet and, if so, give length
bool loop(vector<unsigned long long>& v, unsigned& loopLen)
```

so that it accepts to positive integers as seeds and constructs the `SubPrimeFib` object with all of the data member properly constructed.

Use the struct `Noderun` below to build the `subPrimeFib` and to print out the cycle in the `(a,b)-r-` format as shown further below.

```
struct Noderun {
    unsigned long long n1;
    unsigned long long n2;
    unsigned run;
    Noderun(unsigned long long a, unsigned long long b, unsigned r)
            :n1(a), n2(b), run(r) {}
};
```

Here is a typical console output of the program showing the kind of information to show in the interaction:

```
Enter two seeds for the subprime Fibonacci sequence: 1 1
1 1 2 3 5 4 3 7 5 6 11 17 14 31 15 23 19 21 20 41 61 51 56 107 163 135 149 142 9
7 239 168 37 41 39 40 79 17 48 13 61 37 49 43 46 89 45 67 56 41 97 69 83 76 53 4
3 48 13
The sequence contains 57 numbers
and the cycle length is 18
The nodes are runs of the cycle are:
(13,61)-6-(89,45)-4-(41,97)-5-(53,43)-3-

Enter two seeds for the subprime Fibonacci sequence: 234 4574
234 4574 2404 3489 83 1786 623 803 713 758 1471 743 1107 925 1016 647 1663 1155
1409 1282 897 2179 1538 1239 2777 2008 1595 1201 1398 113 1511 812 101 83 92 35
127 81 104 37 47 42 89 131 110 241 117 179 148 109 257 183 220 31 251 141 196 33
7 41 189 115 152 89 241 165 203 184 129 313 221 267 244 73 317 195 256 41 99 70
13 83 48 131 179 155 167 161 164 65 229 147 188 67 85 76 23 33 28 61 89 75 82 15
7 239 198 23 17 20 37 19 28 47 25 36 61 97 79 88 167 85 126 211 337 274 47 107 7
7 92 13 35 24 59 83 71 77 74 151 75 113 94 69 163 116 93 19 56 25 27 26 53 79 66
 29 19 24 43 67 55 61 58 17 25 21 23 22 15 37 26 21 47 34 27 61 44 35 79 57 68 2
5 31 28 59 29 44 73 39 56 19 25 22 47 23 35 29 32 61 31 46 11 19 15 17 16 11 9 1
0 19 29 24 53 11 32 43 25 34 59 31 45 38 83 11 47 29 38 67 35 51 43 47 45 46 13
59 36 19 11 15 13 14 9 23 16 13 29 21 25 23 24 47 71 59 65 62 127 63 95 79 87 83
 85 84 13 97 55 76 131 69 100 13 113 63 88 151 239 195 217 206 141 347 244 197 1
47 172 29 67 48 23 71 47 59 53 56 109 55 82 137 73 105 89 97 93 95 94 63 157 110
 89 199 144 49 193 121 157 139 148 41 63 52 23 25 24 7 31 19 25
The sequence contains 325 numbers
and the cycle length is 136
The nodes and runs of the cycle are:
(47,23)-5-(61,31)-3-(11,19)-5-(11,9)-3-(19,29)-3-(53,11)-3-(43,25)-3-
(59,31)-4-(83,11)-5-(67,35)-7-(13,59)-3-(19,11)-5-(9,23)-3-(13,29)-6-
(47,71)-5-(127,63)-8-(13,97)-4-(131,69)-3-(13,113)-4-(151,239)-5-
(141,347)-3-(197,147)-3-(29,67)-3-(23,71)-6-(109,55)-3-(137,73)-8-
(63,157)-3-(89,199)-3-(49,193)-6-(41,63)-3-(23,25)-3-(7,31)-5-
Enter two seeds for the subprime Fibonacci sequence:
```

Submit the code using your initials in the usual format: say <Initials>_subFib.cpp