Algorithmic Problems

Andreas Klappenecker



Iterated Functions

Let S be a finite set and f: $S \rightarrow S$ a function mapping S into itself. Form a sequence by choosing an initial value x_0 in S and then $x_{i+1} = f(x_i)$

for all i >= 0. In other words, $x_k = f^k(x_0)$. Show that there must exist indices s and L such that $x_s = x_{s+L} = f^L(x_s)$



Proof

Let S have n elements. By the pigeonhole principle, the n+1 values $x_0, f(x_0), f(f(x_0)), f(f(f(x_0))), ..., f^n(x_0)$ must have at least one repeated value. Thus, there exist indices s and s+L, L>O, such that $x_{s} = f^{s}(x_{0}) = f^{s+L}(x_{0}) = f^{L}(x_{s}).$

When L is minimal, then we call it the cycle length.

Problem

Given a start value x_0 and a function f: S -> S, write an algorithm to find the first s and minimal L such that $x_{s} = f^{s}(x_{0}) = f^{s+L}(x_{0}).$

In other words, find the start s of the cycle and it's cycle length L.

The Hare and the Tortoise

Show that there must exist an index n such that

 $X_n = X_{2n}$



Proof

In the sequence, we get some repeated values $x_n = x_m$ with m>n when m-n is a multiple of the cycle length L and both indices not smaller than s.

Thus, we have $x_{2n} = x_n$ for every index $n \ge s$ such that n is a multiple of the cycle length L.

How large is n?

After s steps, the tortoise enters the cycle. The faster hare is already in the cycle.

Then after less than s+L steps total, the hare and the tortoise meet.

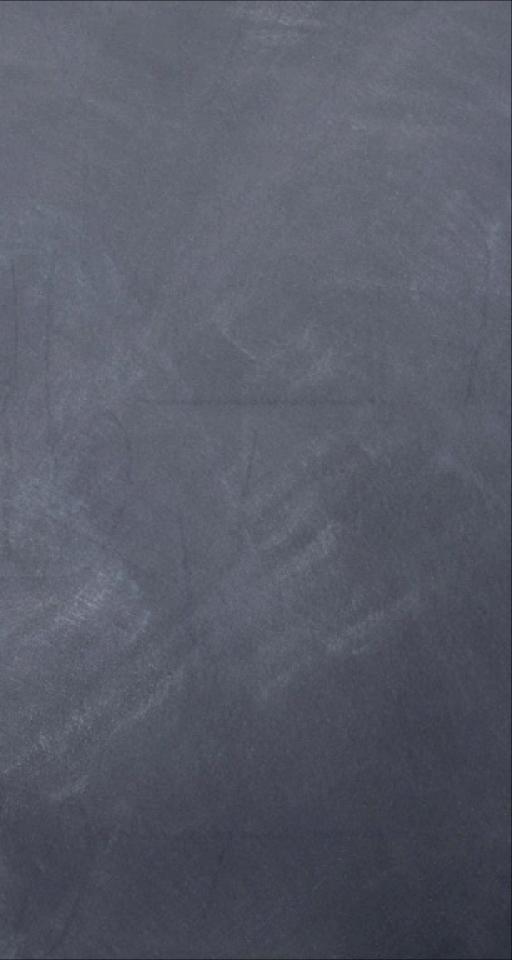
Indeed, the interval [s, s+L-1] contains L different numbers, and one must be a multiple of L. So s <= n = mL < s+L, which yields $m = \lceil s/L \rceil$. Therefore, $n = L \lceil s/L \rceil <= s+L$.

Tortoise and Hare

 $t = x_0; h = x_0; n = 0;$

repeat

t = f(t); h = f²(h); n = n+1; until t=h; return n; // n is a multiple of the cycle length L



Problem

Suppose that we know a multiple n of the cycle length L. How can we find the start s of the cycle and the cycle length L? Find an algorithm that uses O(s+L) steps.

Floyd's Idea

Let n be such that $x_{2n} = x_n$. Find first k such that $f^k(x_0) = f^k(x_n)$, or $x_k = x_{k+n}$. Then we must have s = k, so we have found the start of the cycle. Search for the first index $k \ge s$ such that $x_k = x_s$. Then L = k - s is the length of the cycle.

Time and Space Estimates

Floyd's cycle detection algorithm uses just two variables, so O(1)memory usage!

Finding n can be done in less than s+L steps. Finding the start of the cycle uses additionally s steps. At most L additional steps are needed to find the cycle length L. Total: O(s+L) steps.

Applications

Test the quality of pseudo-random number generators.
Test whether a linked list has a loop
Pollard's rho algorithm for factoring integers

Birthday Paradox and Factoring

Suppose that a number N is the product of two distinct prime p and q, so N=pq. Pick k numbers x_i uniformly at random from the range [2,N-1]. If $gcd(x_i - x_j, n) > 1$, then we have found a factor. Problem: We need to store $k > N^{1/4}$ numbers.

Pollard's Rho Algorithm

```
Let f(x) = x^2 + 1 \mod N
a = 2;
b = 2;
while ( b != a ){
  a = f(a);
  b = f(f(b));
   p = GCD(b - a, N);
  if (p > 1)
     return "Found factor: p";
}
```

return "Fail"

