Math 502 Class Slides

William F. Hammond

Dept. of Mathematics & Statistics
University at Albany
Albany, New York 12222 (USA)
http://www.albany.edu/~hammond/

January 31, 2008

1 Defining the Syracuse Function in Maple

There are two ways to define the Syracuse function in *Maple*:

1. `syra := n -> if n<=1 then 1 elif n mod 2 = 0 then n/2 else 3*n+1 fi;`
2. `syrb := n -> piecewise(n<=1, 1, n mod 2 = 0, n/2, 3*n+1);`

2 Counting Iterates to 1

Use *Maple*'s `proc` facility to write a function csyr that counts the number of iterations of the Syracuse function from a given integer until the number 1 is obtained as an iterate. (For example, csyr(6) = 8.)

```
csyr := proc (n) local c, t;
c := 0;
t := n;
while c < 1000 and 1 < t do
  c := c+1;
  t := syr(t)
  end do;
c
end proc;
```

In this code the number 1000 provides an upper limit on the number of iterations of the *while* loop so that the code cannot possibly run forever. A more flexible procedure definition would make the iteration limit part of the function's calling sequence, e.g., newcsyr(6, 1000) instead of csyr(6). *Maple* provides a variable `"nargs"` for use in a procedure definition to detect the number of arguments used by the caller.

3 Last Assignment Problem

The statement is this:

> Find the 5 smallest values of n for which the first $2n + 1$ iterations of s (the Syracuse function) applied to n fail to yield 1.

Solution. Run the following *for* loop:

```
for j from 1 to 1000 do
  if csyr(j) > 2*j + 1 then
    print(j);
  end if;
  end do;
```

An upper limit such as 1000 is desirable as a means of preventing the loop from running forever. The choice of 1000 as loop size is a blind guess. It may or may not turn out to be large enough. (But for this problem it is large enough.) Upon running this loop one finds 7 values in the sought category, of which the five smallest are $7, 27, 31, 41, 47$.