

Best d Value of El-Sedy and Siksek for Consecutive Happy Numbers

1 Apr 2008

Note that Dr. Grundman uses S_2 but we will simply use S in the explanation, which in our program is the procedure 'onestep'.

Our goal here is to find the minimum ell in the El Sedy and Sistek paper.

First we define our procedures.

```
> restart;  
> randomize( );  
  
1207451400 (1)
```

```
> f := n → n^2; #in case we ever want to investigate the cube of the digits, etc.  
f := n → n^2 (2)
```

```
> bs := 10;  
#this is the base, in case we ever want to investigate binary or ternary or any other base.  
bs := 10 (3)
```

```
> onestep := proc(n1)  
#this is what Dr. Grundman calls S_2(n1) and what we will simply call S below.  
local ans, n, d;  
n := n1;  
ans := 0;  
while n > 0 do  
d := n mod bs;  
ans := ans + f(d);  
n := (n - d) / bs;  
end do;  
ans;  
end;
```

```
onestep := proc(n1) (4)  
local ans, n, d;  
n := n1;  
ans := 0;  
while 0 < n do d := mod(n, bs); ans := ans + f(d); n := (n - d) / bs end do;  
ans  
end proc
```

```
> happy := proc(n)  
# returns -1 if not happy, and returns the number of steps to reach 1 if it is happy  
local m, j, height;  
m := n;  
height := -1;  
for j from 1 to 100 while (m > 1 and m ≠ 4) do  
m := onestep(m);
```

```

    end do;
    if m = 1 then height := j; end if;
    height;
    end;
happy := proc(n)
    local m, j, height;
    m := n;
    height := - 1;
    for j to 100 while 1 < m and m <> 4 do m := onestep(m) end do;
    if m = 1 then height := j end if;
    height
end proc

```

(5)

Our goal here is to find the minimal l value of Sedy and Siksek.

We first consider numbers l ending with digits less than 854 so there is no carry.

```

> Dlist := [1, 4, 16, 37, 58, 89, 145, 42, 20];
           Dlist := [1, 4, 16, 37, 58, 89, 145, 42, 20]
> for m0 from 0 to 854 do
    for m1 from 1 to 1000 do
        fini := false;
        for j from 1 to 9 while fini = false do
            N := m1 + onestep(m0 + Dlist[j]);
            if happy(N) = -1 then fini := true; end if;
        end do;
        if fini = false then print(yes, m0, m1) end if;
    end do;
end do;

```

(6)

so none in this range. Now try for four digit with no carries to fifth.

```

> for m0 from 0 to 9854 do
    for m1 from 1 to 10000 do
        fini := false;
        for j from 1 to 9 while fini = false do
            N := m1 + onestep(m0 + Dlist[j]);
            if happy(N) = -1 then fini := true; end if;
        end do;
        if fini = false then print(m0, m1) end if;
    end do;
end do;

```

292, 9664
348, 6255
525, 3266
725, 3242
1292, 9663
1348, 6254
1525, 3265
1725, 3241

Warning, computation interrupted

```
> for m1 from 1 to 10000 do
  for m0 from 1 to 9854 do
    fini := false;
    for j from 1 to 9 while fini = false do
      N := m1 + onestep(m0 + Dlist[j]);
      if happy(N) = -1 then fini := true; end if;
    end do;
    if fini = false then print(m0, m1) end if;
  end do;
end do:
```

9725, 3161
8725, 3178
9525, 3185
7725, 3193
8525, 3202
6725, 3206
5725, 3217
7525, 3217
4725, 3226
6525, 3230
3725, 3233
2725, 3238
1725, 3241
5525, 3241
725, 3242
4525, 3250
3525, 3257
2525, 3262
1525, 3265
525, 3266
9348, 6174
8348, 6191
7348, 6206
6348, 6219
5348, 6230
4348, 6239
3348, 6246
2348, 6251
1348, 6254
348, 6255
9292, 9583
8292, 9600
7292, 9615

6292, 9628
5292, 9639
4292, 9648
3292, 9655
2292, 9660
1292, 9663

(7)

Now check to see if all but 145 and 89 work.

```
> for m1 from 1 to 3400 do
  for m0 from 9854 to 9999 - 58 do
    fini := false;
    for j from 1 to 7 while fini = false do
      N := m1 + onestep(m0 + Dlist[j]);
      if happy(N) = -1 then fini := true; end if;
    end do;
    if fini = false then print(m0, m1) end if;
  end do;
end do;
```

9880, 3103

(8)

```
> for m1 from 3103 to 3103 do
  for m0 from 9880 to 9880 do
    fini := false;
    for j from 8 to 8 while fini = false do
      N := m1 + onestep(m0 + Dlist[j]);
      if happy(N) = -1 then fini := true; end if;
    end do;
    if fini = false then print(m0, m1) end if;
  end do;
end do;
```

so we do not need to worry about any m0 up to 9941.

```
> for m1 from 1 to 3400 do
  for m0 from 9941 to 9999 do
    for k from 0 to iquo(m1, 81) do
      for d from 0 to 8 do
        fini := false;
        for j from 1 to 9 while fini = false do
          if m0 + Dlist[j] < 10000 then N := m1 + onestep(m0 + Dlist[j]);
          else
            m2 := (m0 + Dlist[j]) mod 10000;
            m3 := m1 - k*81 + (d + 1)2 - d2;
            N := m3 + onestep(m2);
          end if;
          if happy(N) = -1 then fini := true; end if;
        end do;
        if fini = false then print(m0, m1, k, d) end if;
      end do; end do; end do;
    end do;
```

9969, 1267, 8, 0

Warning, computation interrupted

```
> m1
```

1522

(9)

this value leads to $ell = 4699999990999999999969$.

>
>
>
>