Factorisation

May 4, 2022

Contents

Preface

This is my CATAM project, 15.10 for part II. The code for each question can be found in section 3.

1 Factor Bases

1.1 Question 1

See below for a table of estimates for the probability that a d-digit number is B-smooth for $2 \leq d \leq 10$, with B the set of primes less than 50.

Number of Digits	Probability
2	8.7×10^{-1}
3	4.7×10^{-1}
4	2.0×10^{-1}
5	7.2×10^{-2}
6	2.2×10^{-2}
7	6.3×10^{-3}
8	1.6×10^{-3}
9	3.9×10^{-4}
10	1.0×10^{-4}

Table 1: Probabilities for a d-digit number being B-smooth

2 Continued Fractions

2.1 Question 2

We show by induction that the continued fraction of $x = \sqrt{N}$ has each x_n of the form $\frac{r_n + \sqrt{N}}{s_n}$ with $s \mid (r^2 - N)$. Indeed for $n = 0$ this holds with $r_0 = 0, s_0 = 1$. Then suppose it holds for n, then we have

$$\begin{aligned}
x_{n+1} &= \frac{1}{x_n - a_n} \\
&= \frac{s_n}{r_n + \sqrt{N} - a_n s_n} \\
&= \frac{s_n(r_n - s_n a_n - \sqrt{N}}{(r_n - s_n a_n)^2 - N} \\
&= \frac{(s_n a_n - r_n) + \sqrt{N}}{2 r_n a_n - s_n a_n^2 + \frac{N - r_n^2}{s_n}}.
\end{aligned}$$

Then set

$$\begin{aligned}
r_{n+1} &= s_n a_n - r_n \\
s_{n+1} &= 2 r_n a_n - s_n a_n^2 + \frac{N - r_n^2}{s_n}.
\end{aligned}$$

This completes the induction since $(s_n a_n - r_n)^2 - N = s_n s_{n+1}$ and so $s_{n+1} \mid (r_{n+1}^2 - N)$.

See below a table of the first 10 partial quotients of the continued fraction expansion of $\sqrt{N}$ for $1 \leq N \leq 50$.

N	Partial Quotients									
1	1	-	-	-	-	-	-	-	-	-
2	1	2	2	2	2	2	2	2	2	2
3	1	1	2	1	2	1	2	1	2	1
4	2	-	-	-	-	-	-	-	-	-
5	2	4	4	4	4	4	4	4	4	4
6	2	2	4	2	4	2	4	2	4	2
7	2	1	1	1	4	1	1	1	4	1
8	2	1	4	1	4	1	4	1	4	1
9	3	-	-	-	-	-	-	-	-	-
10	3	6	6	6	6	6	6	6	6	6
11	3	3	6	3	6	3	6	3	6	3
12	3	2	6	2	6	2	6	2	6	2
13	3	1	1	1	1	6	1	1	1	1
14	3	1	2	1	6	1	2	1	6	1
15	3	1	6	1	6	1	6	1	6	1
16	4	-	-	-	-	-	-	-	-	-
17	4	8	8	8	8	8	8	8	8	8
18	4	4	8	4	8	4	8	4	8	4
19	4	2	1	3	1	2	8	2	1	3
20	4	2	8	2	8	2	8	2	8	2
21	4	1	1	2	1	1	8	1	1	2
22	4	1	2	4	2	1	8	1	2	4
23	4	1	3	1	8	1	3	1	8	1
24	4	1	8	1	8	1	8	1	8	1
25	5	-	-	-	-	-	-	-	-	-
26	5	10	10	10	10	10	10	10	10	10
27	5	5	10	5	10	5	10	5	10	5
28	5	3	2	3	10	3	2	3	10	3
29	5	2	1	1	2	10	2	1	1	2
30	5	2	10	2	10	2	10	2	10	2
31	5	1	1	3	5	3	1	1	10	1
32	5	1	1	1	10	1	1	1	10	1
33	5	1	2	1	10	1	2	1	10	1
34	5	1	4	1	10	1	4	1	10	1
35	5	1	10	1	10	1	10	1	10	1
36	6	-	-	-	-	-	-	-	-	-
37	6	12	12	12	12	12	12	12	12	12
38	6	6	12	6	12	6	12	6	12	6
39	6	4	12	4	12	4	12	4	12	4
40	6	3	12	3	12	3	12	3	12	3
41	6	2	2	12	2	2	12	2	2	12
42	6	2	12	2	12	2	12	2	12	2
43	6	1	1	3	1	5	1	3	1	1
44	6	1	1	1	2	1	1	1	12	1
45	6	1	2	2	2	1	12	1	2	2
46	6	1	3	1	1	2	6	2	1	1
47	6	1	5	1	12	1	5	1	12	1
48	6	1	12	1	12	1	12	1	12	1
49	7	-	-	-	-	-	-	-	-	-
50	7	14	14	14	14	14	14	14	14	14

Table 2: Partial Quotients of CF Expansion of various $\sqrt{N}$

2.2 Question 3

See below a table of the $P_n^2 - NQ_n^2$ for $1 \geq n \geq 10$, and N ranging over non-square integers in $[1, 50]$. Note that the equation $x^2 - Ny^2 = -1$ has no solution for $N \leq 0$ since then $x^2 - Ny^2 \geq 0 > -1$ for $x, y \in \mathbb{Z}$.

We can see from the table of $P_n^2 - NQ_n^2$ that we can generate many solutions of $x^2 - Ny^2 = \pm 1$ by finding P_n, Q_n and checking if $P_n^2 - NQ_n^2 = \pm 1$. In fact it is a general fact of number theory (proved in the part II C course Number Theory) that (P_{kn-1}, Q_{kn-1}) is a solution of $x^2 - Ny^2$ for all k such that kn is even. So we can be sure this a good way of generating solutions to Pell's Equation.

$N \backslash n$	$P_n^2 - NQ_n^2$									
	1	2	3	4	5	6	7	8	9	10
2	-1	1	-1	1	-1	1	-1	1	-1	1
3	-2	1	-2	1	-2	1	-2	1	-2	1
5	-1	1	-1	1	-1	1	-1	1	-1	1
6	-2	1	-2	1	-2	1	-2	1	-2	1
7	-3	2	-3	1	-3	2	-3	1	-3	2
8	-4	1	-4	1	-4	1	-4	1	-4	1
10	-1	1	-1	1	-1	1	-1	1	-1	1
11	-2	1	-2	1	-2	1	-2	1	-2	1
12	-3	1	-3	1	-3	1	-3	1	-3	1
13	-4	3	-3	4	-1	4	-3	3	-4	1
14	-5	2	-5	1	-5	2	-5	1	-5	2
15	-6	1	-6	1	-6	1	-6	1	-6	1
17	-1	1	-1	1	-1	1	-1	1	-1	-
18	-2	1	-2	1	-2	1	-2	1	-2	1
19	-3	5	-2	5	-3	1	-3	5	-2	5
20	-4	1	-4	1	-4	1	-4	1	-4	1
21	-5	4	-3	4	-5	1	-5	4	-3	4
22	-6	3	-2	3	-6	1	-6	3	-2	3
23	-7	2	-7	1	-7	2	-7	1	-7	2
24	-8	1	-8	1	-8	1	-8	1	-8	1
26	-1	1	-1	1	-1	1	-1	1	0	0
27	-2	1	-2	1	-2	1	-2	1	-2	0
28	-3	4	-3	1	-3	4	-3	1	-3	4
29	-4	5	-5	4	-1	4	-5	5	-4	1
30	-5	1	-5	1	-5	1	-5	1	-5	1
31	-6	5	-3	2	-3	5	-6	1	-6	5
32	-7	4	-7	1	-7	4	-7	1	-7	4
33	-8	3	-8	1	-8	3	-8	1	-8	3
34	-9	2	-9	1	-9	2	-9	1	-9	2
35	-10	1	-10	1	-10	1	-10	1	-10	1
37	-1	1	-1	1	-1	1	-1	0	0	131072
38	-2	1	-2	1	-2	1	-2	1	-4	0
39	-3	1	-3	1	-3	1	-3	1	-3	0
40	-4	1	-4	1	-4	1	-4	1	-4	1
41	-5	5	-1	5	-5	1	-5	5	-1	5
42	-6	1	-6	1	-6	1	-6	1	-6	1
43	-7	6	-3	9	-2	9	-3	6	-7	1
44	-8	5	-7	4	-7	5	-8	1	-8	5
45	-9	4	-5	4	-9	1	-9	4	-5	4
46	-10	3	-7	6	-5	2	-5	6	-7	3
47	-11	2	-11	1	-11	2	-11	1	-11	2
48	-12	1	-12	1	-12	1	-12	1	-12	1
50	-1	1	-1	1	-1	1	-1	0	0	0

Table 3: Values of $P_n^2 - NQ_n^2$ for various n, N

3 Code

3.1 Question 1

```
function [outputArg1,outputArg2] = Bsmooth(N,B)
i=1;
k=zeros(size(B,2),2);
while i<= size(B,2)
    p=B(i);
    if mod(N,p)==0
        k(i)=k(i)+1;
        N=N/p;
    else
        i=i+1;
    end
end
i=1;
while i<= size(B,2)
NN=prod(B(i)^k(i));
i=i+1;
end
diff=N-NN;
if diff ==0
    disp(['N is B smooth'])
else
    disp(['N is not B smooth'])
end



end
```

3.2 Question 2

```
function [outputArg1,outputArg2] = cfsqrt(N,max)

x(1)=sqrt(N);
a(1)=floor(x(1));
r(1)=0;
s(1)=1;

i=2;

while i<= max
    r(i)=-(r(i-1)-(s(i-1)*a(i-1)));
    s(i)=-(((r(i-1)^2-N)/s(i-1))-(2*r(i-1)*a(i-1))+(s(i-1)*(a(i-1)^2)));
    x(i)=(r(i)+sqrt(N))/s(i);
    a(i)=floor(x(i));
    i=i+1;
end
a


```

```
end
```

3.3 Question 6

```
function [outputArg1,outputArg2] = F2Gauss(A)

length=size(A,1);
width=size(A,2);



i=1;
j=1;
while i<= length && j<= width

finish =0;

while  finish ==0

    if norm(A([i:length],j))==0
            j=j+1;
            if j> width
                finish=1;
            end
    else
                A(i,j);
            %Rearrange rows
                if A(i,j)== 1

                else

                    k=i+1;
                    fin=0;
                    while fin==0 && k<= length
                        if A(k,j) == 1
                            hold=A(i,:);
                            A(i,:)=A(k,:);
                            A(k,:)=hold;
                            fin=1;
                        else
                            k=k+1;
                        end
                    end
                end
                i;
                j;
                A;

```

```
                    %subtract away bad rows

                    k=i+1;
                    while k<= length

                        if A(k,j) == 0
                        else
                            A(k,:)=mod(A(k,:)-A(i,:),2);
                        end
                        k=k+1;
                    end
                    finish=1;
                    A;
    end
end
i=i+1;
j=j+1;
end
A;
decided=zeros(width,1);

if width <= length && A(width,width)==1
    disp(['Kernel is trivial'])
    v=zeros(width,1);
elseif norm(A)==0
    v=ones(width,1);
else
    v=zeros(width,1);
    %find first row not all zero
    term=0;
    i=length;
    while term==0
        if norm(A(i,:))==0
            i=i-1;
        else
            st=i;
            term=1;
        end
    end
    %sort out any relationship on first row
    i=st;

    while i >= 1
        j=1;
        term=0;
        while term ==0
            if A(i,j)==1
                term=1;
            else
                j=j+1;
            end
        end
         jj=j+1;
        while jj<= width
```

```
                if decided(jj)==0
                v(jj)=1;
                decided(jj)=1;
                end
                jj=jj+1;
            end
            if j==width || norm(A(i,[j+1:width]))==0
                v(j)=0;
                decided(j)=1;
            else
                x=A(i,[j+1:width]).';
                y=v([j+1:width]);
                z=x.*y;
                v(j)=mod(-sum(z),2);
                decided(j)=1;
            end
            i=i-1;

    end
end
A
v
mod(A*v,2)








end
```