Section 7 $[\zeta(s)^z]_n$:

---

7. Computing $\Pi(n)$ with Another Combinatorial Approach

Taking inspiration from an algorithm published by Marc Deléglise and Joöl Rivat in their paper "Computing the summation of the Mobius function" (see http://projecteuclid.org/download/pdf_1/euclid.em/1047565447) this section details a combinatorial method for computing $[\zeta(0)^z]_n$ and $\Pi(n)$ in $O(n^{2/3}\log n)$ time and $O(n^{1/3}\log n)$ space. That paper makes a handy reference for what follows. This section is particularly hard to follow, owing to its complexity.

7.1 Defining $[f^k]_n$

Suppose, for some function $f(n)$, we have the following summatory functions

$$[f^k]_n = \sum_{j=1}^{\lfloor n \rfloor} f(j)[f^{k-1}]_{n\, j^{-1}}$$
$$\nabla [f^k]_n = [f^k]_n - [f^k]_n \tag{7.1.1}$$

```
F[ fn_, n_, 0] := UnitStep[n-1]; F[ fn_, n_, k_] := Sum[ fn[j] F[fn, n/j, k-1],{j,1,Floor[n]}]
f[ fn_, n_, k_] := F[ fn,n,k]-F[fn,n-1,k]
```

7.2 A Combinatorial Identity for $[f^k]_n$

Then the following rather complicated combinatorial identity holds for $[f^k]_n$, with $1 < t < n$:

$$\boxed{\begin{aligned}&[f^k]_n = [f^k]_t + \\ &\quad\sum_{j=t+1}^{\lfloor n \rfloor} f(j)\cdot[f^{k-1}]_{n\cdot j^{-1}} \\ &\quad+\sum_{j=1}^{t}\ \sum_{s=\lfloor t\cdot j^{-1}\rfloor+1}^{\lfloor n\cdot j^{-1}\rfloor}\ \sum_{m=1}^{k-1} f(s)\cdot\nabla[f^m]_j\cdot[f^{k-m-1}]_{n(j\,s)^{-1}}\end{aligned}} \tag{7.2.1}$$

```
F[fn_,n_,0]:=UnitStep[n-1]
F[fn_,n_,k_]:=F[fn,n,k]=Sum[fn[j] F[fn,n/j,k-1],{j,1,Floor[n]}]
f[fn_,n_,k_]:=F[fn,n,k]-F[fn,n-1,k]
FAlt[fn_,n_,k_,t_]:=F[fn, t,k]+Sum[fn[j] F[fn,n/j,k-1],{j,t+1,Floor[n]}]+Sum[fn[s]f[fn,j,m]F[fn,n/(j s),k-m-1],{j,1,t},
{s,Floor[t/j]+1,Floor[n/j]},{m,1,k-1}]
id[n_] := 1
Grid[Table[F[id, n,k]-FAlt[id, n,k, Floor[ n^(1/2)]],{n,10,500,10},{k,1,7}]]
```

```
Grid[Table[F[MoebiusMu, n,k]-FAlt[MoebiusMu, n,k, Floor[ n^(1/2)]],{n,10,500,10},{k,1,7}]]
```

Inspection shows that, in (6.2), the largest argument for $[f^k]_n$ is $\frac{n}{t}$, and the largest for $\nabla[f^k]_n$ is t . The only exceptions are for $[f]_n$, where the largest argument is *n,* and $f(n)$, which also takes arguments up to *n*.

The reason this identity is useful is that, if $f(n)$ and $[f]_n=\sum_{j=1}^{n} f(j)$ can be computed in constant time, and if we have some external method to compute a table of $[f^k]_n$ up to arguments of $\frac{n}{t}$ and $\nabla[f^k]_n$ up to arguments of t , then we can use (6.2) to compute $[f^k]_n$.

7.3 Reducing Computation of Certain Sums from n Steps to $2n^{\frac{1}{2}}$ Steps

Our goal is to compute $[f^k]_n$ with (6.2) as efficiently as possible, so we need another important identity. Inspection of $[f^k]_n$ in (6.1) should make clear that $[f^k]_n=[f^k]_{\lfloor n\rfloor}$. Now, it's the case that if we have functions $g(n)$ and $h(n)$ such that $g(n)=g(\lfloor n\rfloor)$ and $h(n)=h(\lfloor n\rfloor)$, then the sum $\sum_{j=1}^{\lfloor n\rfloor}(g(j)-g(j-1))h(\frac{n}{j})$ can be split into two parts as

$$\sum_{j=1}^{\lfloor n\rfloor}(g(j)-g(j-1))h(\frac{n}{j})=\sum_{j=1}^{\lfloor n^{\frac{1}{2}}\rfloor}(g(j)-g(j-1))h(n j^{-1})+\sum_{j=1}^{\lfloor n\cdot\lfloor n^{\frac{1}{2}}\rfloor^{-1}\rfloor-1}(g(n j^{-1})-g(n(j+1)^{-1}))\cdot h(j) \tag{7.3.1}$$

```
s1[ n_, g_, h_ ] := Sum[ (g[j]-g[j-1]) h[ n/j],{j,1,Floor[n]}]
s2[ n_, g_, h_ ] := Sum[ (g[j]-g[j-1]) h[ n/j], {j, 1, Floor[ n^(1/2)]}]+Sum[ (g[n/j]-g[n/(j+1)])h[j],
{j,1,Floor[ n/Floor[( n^(1/2))]]-1}]
id[ n_ ] := Floor[n]
mert[n_] := Sum[ MoebiusMu[ j ], {j,1,Floor[n]}]
Table[ { s1[ n, id, id ],  "=", s2[n, id, id], "  ", s1[ n,mert,mert ],  "=", s2[n, mert,mert] }, {n,100,1000,100}]// TableForm
```

7.4 A More Efficient Variant of (6.2)

Variants of (6.3) can be applied to two of the sums in (6.2), as long as *t* is less than $n^{\frac{1}{2}}$, to leave it as

$$\begin{aligned}[f^k]_n=[f^k]_t+&\\ &\sum_{j=t+1}^{\lfloor n^{\frac{1}{2}}\rfloor} f(j)\cdot[f^{k-1}]_{n\cdot j^{-1}}\\ &+\sum_{j=1}^{\lfloor n\cdot\lfloor x^{\frac{1}{2}}\rfloor^{-1}-1\rfloor}([f]_{n\cdot j^{-1}}-[f]_{n\cdot(j+1)^{-1}})\cdot\nabla[f^{k-1}]_j\\ &+\sum_{j=1}^{t}\sum_{s=\lfloor t\cdot j^{-1}+1\rfloor}^{\lfloor\lfloor n\cdot j^{-1}\rfloor^{\frac{1}{2}}\rfloor}\sum_{m=1}^{k-1} f(s)\cdot\nabla[f^m]_j\cdot[f^{k-m-1}]_{n\cdot(js)^{-1}}\\ &+\sum_{j=1}^{t}\sum_{s=1}^{\lfloor\lfloor n\cdot j^{-1}\rfloor\cdot\lfloor\lfloor n\cdot j^{-1}\rfloor^{\frac{1}{2}}\rfloor^{-1}-1\rfloor}([f]_{n(js)^{-1}}-[f]_{n\cdot(j(s+1))^{-1}})\cdot\nabla[f^m]_j\cdot[f^{k-m-1}]_s\end{aligned} \tag{7.4.1}$$

```
F[fn_,n_,k_, s_]:=F[fn,n,k,s]=Sum[(fn[m]^(k-j)) Binomial[k,j]F[ fn,n/(m^(k-j)),j, m+1],{m,s,n^(1/k)},{j,0,k-1}]
F[fn_,n_,0, s_]:=UnitStep[n-1]
```

```
F[fn_,n_,k_]:= F[fn,n,k,1]
f[fn_,n_, k_]:=F[fn,n,k]-F[fn,n-1,k]
FAlt[fn_, n_, k_, t_]:=F[fn,t, k]+Sum[fn[j] F[fn, n/j, k-1],{j,t+1,n^(1/2)}]+
  Sum[Sum[fn[m],{m,Floor[n/(j+1)]+1,n/j}]F[fn, j, k-1],{j,1,n/Floor[n^(1/2)]-1}]+
  Sum[fn[s] f[fn, j, m] F[fn,n/(j s),k-m-1],{j,1,t},{s,Floor[t/j]+1,Floor[n/j]^(1/2)},{m,1,k-1}]+Sum[(Sum[fn[m],{m,Floor[n/(j(s+1))]+1,n/(j s)}])(Sum[f[fn,j, m] F[fn,s, k-m-1],{m,1,k-1}]),{j,1,t},{s,1,Floor[n/j]/Floor[Floor[n/j]^(1/2)]-1}]
FAlt[fn_, n_, 1, t_]:=Sum[fn[j],{j,1,n}]
Grid[Table[F[MoebiusMu, n,k,1]-FAlt[MoebiusMu, n,k,Floor[n^(1/3)]],{n,10,500,10},{k,1,7}]]
Grid[Table[F[LiouvilleLambda, n,k,1]-FAlt[LiouvilleLambda, n,k,Floor[n^(1/3)]],{n,10,500,10},{k,1,7}]]
```

7.5 Applying the Preceding Techniques to Compute $[(\zeta(0)-1)^k]_n$

Now let's define our function $f(n)$ and choose a value for *t*.

Our value for *t* will be $n^{\frac{1}{3}}$.

Our function $f(n)$ will be $f(n)=0\, if\ n=1, 1\, otherwise$. This will satisfy our requirement that $f(n)$ be computable for any value of *n* in constant time.

Thus, our function $[f]_n$ will be $[\zeta(0)-1]_n=\lfloor n\rfloor-1$, also computable for any *n* in constant time.

Our function $[f^k]_n$ will be $[(\zeta(0)-1)^k]_n$, defined in (1.4).

And our function $\nabla[f^k]_n=[f^k]_n-[f^k]_{n-1}$, which is just $[(\zeta(0)-1)^k]_n-[(\zeta(0)-1)^k]_{n-1}$ can also be defined as

$$\nabla[(\zeta(0)-1)^k]_n=\sum_{j|n}\nabla[(\zeta(0)-1)^{k-1}]_j\cdot\nabla[\zeta(0)-1]_{n\cdot j^{-1}}$$
$$\nabla[\zeta(0)-1]_n=1\, if\ n>1, 0\, otherwise$$
$$\nabla[(\zeta(0)-1)^0]_n=1\, if\ n=1, 0\, otherwise \tag{6.5}$$

```
dm1[ n_, k_ ] := Sum[ dm1[ j, k-1 ] dm1[ n/j, 1 ], { j, Divisors[ n ] } ];dm1[ n_, 1 ] := If[ n>1, 1,0];dm1[ n_, 0 ] := 0;dm1[ 1, 0 ] := 1
Grid[ Table[ dm1[ n, k ], { n, 1, 50 }, { k, 1, 7 } ] ]
```

Applying this all to (6.4), we have

$$\boxed{\begin{aligned}[(\zeta(0)-1)^k]_n=&[(\zeta(0)-1)^k]_t+\\ &\sum_{j=\lfloor n^{\frac{1}{3}}\rfloor+1}^{\lfloor n^{\frac{1}{2}}\rfloor}[(\zeta(0)-1)^{k-1}]_{n\cdot j^{-1}}\\ &+\sum_{j=1}^{\lfloor n\cdot\lfloor n^{\frac{1}{2}}\rfloor^{-1}-1\rfloor}(\lfloor n\cdot j^{-1}\rfloor-\lfloor n\cdot(j+1)^{-1}\rfloor)\cdot[(\zeta(0)-1)^{k-1}]_j\\ &+\sum_{j=2}^{\lfloor n^{\frac{1}{3}}\rfloor}\sum_{s=\lfloor\lfloor n^{\frac{1}{3}}\rfloor\cdot j^{-1}+1\rfloor}^{\lfloor\lfloor n\cdot j^{-1}\rfloor^{\frac{1}{2}}\rfloor}\sum_{m=1}^{k-1}\nabla[(\zeta(0)-1)^m]_j\cdot[(\zeta(0)-1)^{k-m-1}]_{n\cdot(js)^{-1}}\\ &+\sum_{j=2}^{\lfloor n^{\frac{1}{3}}\rfloor}\sum_{s=1}^{\lfloor\lfloor n\cdot j^{-1}\rfloor\cdot\lfloor\lfloor n\cdot j^{-1}\rfloor^{\frac{1}{2}}\rfloor^{-1}-1\rfloor}(\lfloor n\cdot(js)^{-1}\rfloor-\lfloor n\cdot(j(s+1))^{-1}\rfloor)\cdot\sum_{m=1}^{k-1}\nabla[(\zeta(0)-1)^m]_j\cdot[(\zeta(0)-1)^{k-m-1}]_s\end{aligned}} \tag{6.6}$$

```
 Dm1[n_,k_]:=Dm1[n,k]=Sum[Dm1[n/j,k-1],{j,2,Floor[n]}];Dm1[n_,0]:=UnitStep[n-1]
dm1[n_,k_]:=Dm1[n,k]-Dm1[n-1,k]
Dm1Alt[n_,k_]:=Dm1[n^(1/3),k]+Sum[Dm1[n/j,k-1],{j,Floor[n^(1/3)]+1,n^(1/2)}]+Sum[(Floor[n/j]-Floor[n/(j+1)])Dm1[j,k-1],{j,1,n/Floor[n^(1/2)]-1}]+Sum[dm1[j,m] Dm1[n/(j s),k-m-1],{j,2,n^(1/3)},{s,Floor[Floor[n^(1/3)]/j]+1,Floor[n/j]^(1/2)},
```

```
{m,1,k-1}]+Sum[(Floor[n/(j s)]-Floor[n/(j(s+1))])(Sum[dm1[j,m] D2[s,k-m-1],{m,1,k-1}]),{j,2,n^(1/3)},
{s,1,Floor[n/j]/Floor[Floor[n/j]^(1/2)]-1}]
Dm1Alt[n_,1]:=Floor[n]-1
Grid[Table[Dm1[n,k]-Dm1Alt[n,k],{n,10,500,10},{k,1,7}]]
```

It is hopefully not too much of a stretch to suggest that if we already had a table with all values for $\nabla[(\zeta(0)-1)^j]_n$ up to arguments of $n^{\frac{1}{3}}$, for $2\le j\le k$, and if we had already had a table with all values of $[(\zeta(0)-1)^j]_n$ up to arguments of $n^{\frac{2}{3}}$, for $2\le j\le k$, and taking into account that $[(\zeta(0)-1)^j]_n=0$ when $n<2^k$, that (6.6) should be able to compute $[(\zeta(0)-1)^j]_n$ for any k in something like $O(n^{\frac{2}{3}}\log n)$ time complexity.

7.6 Sieving

So how would we compute such a table, with values of $\nabla[(\zeta(0)-1)^j]_n$ up to arguments of $n^{\frac{1}{3}}$, and values of $[(\zeta(0)-1)^j]_n$ up to arguments of $n^{\frac{2}{3}}$?

Well, suppose we had a number in prime factored form, $n=\prod_{p^a|n} p^a$. Then we can express $\nabla[\zeta(0)^z]_n$, the function from (1.3), as

$$\nabla[\zeta(0)^z]_n=d_z(n)=\prod_{p^a|n}\frac{z^{(a)}}{a!} \tag{6.7}$$

```
dz[ n_, z_ ] := Product[ (-1)^p[ [ 2 ] ] Binomial[ -z, p[ [ 2 ] ] ], { p, FI[ n ] } ];FI[ n_ ] := FactorInteger[ n ];FI[ 1 ] := { }
 Grid[Table[dz[n,k],{n,1,50},{k,1,7}]]
```

and $\nabla[(\zeta(0)-1)^k]_n$, from (6.5), in terms of $\nabla[\zeta(0)^z]_n$ as

$$\nabla[(\zeta(0)-1)^k]_n=\sum_{j=0}^{k}(-1)^j\binom{k}{j}\nabla[\zeta(0)^{k-j}]_n \tag{6.8}$$

```
dm1[n_,k_]:=Sum[dm1[j,k-1] dm1[n/j,1],{j,Divisors[n]}];dm1[n_,1]:=If[n>1,1,0];dm1[n_,0]:=0;dm1[1,0]:=UnitStep[n-1]
dz[n_,z_]:=Product[(-1)^p[[2]] Binomial[-z,p[[2]]],{p,FI[n]}];FI[n_]:=FactorInteger[n];FI[1]:={}
dm1Alt[n_, k_] := Sum[ (-1)^j Binomial[k,j] dz[ n, k-j], {j,0, k}]
Grid[Table[ dm1[n,k]-dm1Alt[n,k],{n,1,50},{k,1,7}]]
```

and using $\nabla[(\zeta(0)-1)^k]_n$, we can express $[(\zeta(0)-1)^k]_n$ as

$$[(\zeta(0)-1)^k]_n=[(\zeta(0)-1)^k]_{n-1}+\nabla[(\zeta(0)-1)^k]_n \tag{6.9}$$

```
Dm1[ n_, k_ ] := Sum[ Dm1[ n/j, k-1 ], { j, 2, Floor[ n ] } ];Dm1[ n_, 0 ] := UnitStep[n-1]
dz[n_,z_]:=Product[(-1)^p[[2]] Binomial[-z,p[[2]]],{p,FI[n]}];FI[n_]:=FactorInteger[n];FI[1]:={}
dm1[n_, k_] := Sum[ (-1)^j Binomial[k,j] dz[ n, k-j], {j,0, k}]
Dm1Alt[ n_, k_ ] := Dm1[ n-1, k ] + dm1[ n, k ]
Grid[Table[ Dm1[n,k]- Dm1Alt[n,k],{n,1,50},{k,1,7}]]
```

All put together it would look something like this,

```
dz[ n_, z_ ] := Product[ (-1)^p[ [ 2 ] ] Binomial[ -z, p[ [ 2 ] ] ], { p, FI[ n ] } ];FI[ n_ ] := FactorInteger[ n ];FI[ 1 ] := { }
dm1[n_, k_] := Sum[ (-1)^j Binomial[k,j] dz[ n, k-j], {j,0, k}]
Dm1[ n_, k_ ] := Dm1[n,k]=Dm1[ n-1, k ] + dm1[ n, k ]; Dm1[0,k_]:=0
```

So, if we had some way to get numbers in prime factored form and then applied that process sequentially from 1 to $n^{\frac{2}{3}}$, we could use (6.6), (6.7), and (6.8) to build a table of values of $[(\zeta(0)-1)^k]_n$ up to arguments of $n^{\frac{2}{3}}$.

And in fact, we can use a suitable variant of the Sieve of Eratosthenes to do just that.

All told, with sieving and the above three identities, we can compute $[(\zeta(0)-1)^j]_n$ for $2 \le j \le k$ for arguments from 1 to $n^{\frac{2}{3}}$ in something like $O(n^{2/3}\log n)$ time and $O(n^{2/3}\log n)$ space.

We can improve our performance bound to $O(n^{1/3}\log n)$ space if we use a segmented sieve and re-order the way that (6.6) is calculated so that values of $[(\zeta(0)-1)^k]_n$ are applied from smallest arguments to largest, with that application interleaved with sieving of blocks of size $n^{\frac{1}{3}}$.

7.7 Notes and Implementations of the Ideas in This Section

The identity (6.6), coupled with sieving, computes $[(\zeta(0)-1)^k]_n$. We can then use our identity (D1),

$$[\zeta(0)^z]_n = \sum_{k=0}^{\lfloor \log_2 n \rfloor} \binom{z}{k} [(\zeta(0)-1)^k]_n$$

to compute the generalized divisor function $[\zeta(0)^z]_n$ for any *z*, or our identity (P3) to compute the Riemann Prime counting function as

$$\Pi(n) = \sum_{k=1}^{\lfloor \log_2 n \rfloor} \frac{(-1)^{k+1}}{k} [(\zeta(0)-1)^k]_n$$

Computing $[(\zeta(0)-1)^j]_n$ for $2 \le j \le k$ at once, in bulk, presents opportunities for caching and simplification. Thus, the above process can be used to compute $[\zeta(0)^z]_n$ for any *z*, as well as $\Pi(n)$, in something like $O(n^{2/3}\log n)$ time and $O(n^{1/3}\log n)$ space.

A C implementation of this algorithm, being used to count primes in the advertised time and space bounds of $O(n^{2/3}\log n)$ time and $O(n^{1/3}\log n)$ space can be found at *http://www.icecreambreakfast.com/primecount/primescode.html*. Further descriptions of this technique, with better justifications of the combinatorial identities, can be found in *http://www.icecreambreakfast.com/primecount/PrimeCounting_NathanMcKenzie.pdf*