

CONJECTURES ABOUT MIN/MAX PRIME DIVISOR OF THE NATURAL NUMBERS AND PRIME NUMBERS

Edoardo GUEGLIO
egueglio@gmail.com

Applying alternatively maxrad/minrad to the sequence of natural numbers and taking the differences of this sequence you can split this in subsequences by the sum equals length rule so that you can associate each subsequence to each gap between consecutive prime numbers

Premise

All this research has been done using Mathematica©, a symbolic computation environment that uses a programming language called Wolfram Language.

Some useful function

Define rad(n) as the product of all prime numbers that are divisors of n. Now define minrad(n) as the lower of such prime numbers and maxrad(n) the greater of such prime numbers:

```
In[191]= rad[n_Integer]:=Times@@(First/@FactorInteger[n])  
maxrad[n_Integer]:=Max[First/@FactorInteger[n]]  
minrad[n_Integer]:=Min[First/@FactorInteger[n]]
```

Above function FactorInteger(n) returns a list of couples. Each couple is done by a prime number p and an exponent number e where p^e is a divisor of n but not $p^{(e+1)}$. As an example:

```
In[183]= FactorInteger[60]
```

```
Out[183]= {{2, 2}, {3, 1}, {5, 1}}
```

Now apply minrad and maxrad to the natural number starting from 2:

In[184]=

```
minrad/@Range[2,200]
maxrad/@Range[2,200]
```

```
Out[184]= {2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, 3, 2, 17, 2, 19, 2, 3, 2, 23, 2, 5, 2, 3, 2, 29, 2, 31, 2,
3, 2, 5, 2, 37, 2, 3, 2, 41, 2, 43, 2, 3, 2, 47, 2, 7, 2, 3, 2, 53, 2, 5, 2, 3, 2, 59, 2, 61,
2, 3, 2, 5, 2, 67, 2, 3, 2, 71, 2, 73, 2, 3, 2, 7, 2, 79, 2, 3, 2, 83, 2, 5, 2, 3, 2, 89, 2,
7, 2, 3, 2, 5, 2, 97, 2, 3, 2, 101, 2, 103, 2, 3, 2, 107, 2, 109, 2, 3, 2, 113, 2, 5, 2, 3, 2,
7, 2, 11, 2, 3, 2, 5, 2, 127, 2, 3, 2, 131, 2, 7, 2, 3, 2, 137, 2, 139, 2, 3, 2, 11, 2, 5, 2,
3, 2, 149, 2, 151, 2, 3, 2, 5, 2, 157, 2, 3, 2, 7, 2, 163, 2, 3, 2, 167, 2, 13, 2, 3, 2, 173,
2, 5, 2, 3, 2, 179, 2, 181, 2, 3, 2, 5, 2, 11, 2, 3, 2, 191, 2, 193, 2, 3, 2, 197, 2, 199, 2}
```

```
Out[185]= {2, 3, 2, 5, 3, 7, 2, 3, 5, 11, 3, 13, 7, 5, 2, 17, 3, 19, 5, 7, 11, 23, 3, 5, 13, 3, 7, 29, 5, 31, 2,
11, 17, 7, 3, 37, 19, 13, 5, 41, 7, 43, 11, 5, 23, 47, 3, 7, 5, 17, 13, 53, 3, 11, 7, 19, 29,
59, 5, 61, 31, 7, 2, 13, 11, 67, 17, 23, 7, 71, 3, 73, 37, 5, 19, 11, 13, 79, 5, 3, 41, 83,
7, 17, 43, 29, 11, 89, 5, 13, 23, 31, 47, 19, 3, 97, 7, 11, 5, 101, 17, 103, 13, 7, 53, 107,
3, 109, 11, 37, 7, 113, 19, 23, 29, 13, 59, 17, 5, 11, 61, 41, 31, 5, 7, 127, 2, 43, 13, 131,
11, 19, 67, 5, 17, 137, 23, 139, 7, 47, 71, 13, 3, 29, 73, 7, 37, 149, 5, 151, 19, 17, 11,
31, 13, 157, 79, 53, 5, 23, 3, 163, 41, 11, 83, 167, 7, 13, 17, 19, 43, 173, 29, 7, 11, 59,
89, 179, 5, 181, 13, 61, 23, 37, 31, 17, 47, 7, 19, 191, 3, 193, 97, 13, 7, 197, 11, 199, 5}
```

These sequences appear quite unpredictable but more can be conjectured on the difference sequence. The operator difference simply substitute an item in the sequence with the difference between the following item and the item itself for every item; it's like a derivative of the sequence.

In[186]=

```
s1=Differences[minrad/@Range[2,200]]
s2=Differences[maxrad/@Range[2,200]]
```

```
Out[186]= {1, -1, 3, -3, 5, -5, 1, -1, 9, -9, 11, -11, 1, -1, 15, -15, 17, -17, 1, -1, 21, -21, 3, -3, 1,
-1, 27, -27, 29, -29, 1, -1, 3, -3, 35, -35, 1, -1, 39, -39, 41, -41, 1, -1, 45, -45, 5,
-5, 1, -1, 51, -51, 3, -3, 1, -1, 57, -57, 59, -59, 1, -1, 3, -3, 65, -65, 1, -1, 69, -69,
71, -71, 1, -1, 5, -5, 77, -77, 1, -1, 81, -81, 3, -3, 1, -1, 87, -87, 5, -5, 1, -1, 3, -3,
95, -95, 1, -1, 99, -99, 101, -101, 1, -1, 105, -105, 107, -107, 1, -1, 111, -111, 3, -3,
1, -1, 5, -5, 9, -9, 1, -1, 3, -3, 125, -125, 1, -1, 129, -129, 5, -5, 1, -1, 135, -135,
137, -137, 1, -1, 9, -9, 3, -3, 1, -1, 147, -147, 149, -149, 1, -1, 3, -3, 155, -155, 1,
-1, 5, -5, 161, -161, 1, -1, 165, -165, 11, -11, 1, -1, 171, -171, 3, -3, 1, -1, 177, -177,
179, -179, 1, -1, 3, -3, 9, -9, 1, -1, 189, -189, 191, -191, 1, -1, 195, -195, 197, -197}
```

```
Out[187]= {1, -1, 3, -2, 4, -5, 1, 2, 6, -8, 10, -6, -2, -3, 15, -14, 16, -14, 2, 4, 12, -20, 2, 8,
-10, 4, 22, -24, 26, -29, 9, 6, -10, -4, 34, -18, -6, -8, 36, -34, 36, -32, -6, 18, 24,
-44, 4, -2, 12, -4, 40, -50, 8, -4, 12, 10, 30, -54, 56, -30, -24, -5, 11, -2, 56, -50,
6, -16, 64, -68, 70, -36, -32, 14, -8, 2, 66, -74, -2, 38, 42, -76, 10, 26, -14, -18,
78, -84, 8, 10, 8, 16, -28, -16, 94, -90, 4, -6, 96, -84, 86, -90, -6, 46, 54, -104,
106, -98, 26, -30, 106, -94, 4, 6, -16, 46, -42, -12, 6, 50, -20, -10, -26, 2, 120,
-125, 41, -30, 118, -120, 8, 48, -62, 12, 120, -114, 116, -132, 40, 24, -58, -10, 26,
44, -66, 30, 112, -144, 146, -132, -2, -6, 20, -18, 144, -78, -26, -48, 18, -20, 160,
-122, -30, 72, 84, -160, 6, 4, 2, 24, 130, -144, -22, 4, 48, 30, 90, -174, 176, -168,
48, -38, 14, -6, -14, 30, -40, 12, 172, -188, 190, -96, -84, -6, 190, -186, 188, -194}
```

At the first sight it seems that this step added more variability with the introduction of negative items. Moreover these two sequences start with same three terms but after that are completely different. Now apply a simple rule to each sequence. **Collect items in a subsequence until the sum of items equals its length so write it and restart with a new one.** This algorithm can be described with this function:

```
In[188]:= splitseq[l_List] := Module[{i, parz = {}, res = {}},
  For[i = 1, i <= Length[l], i++,
    AppendTo[parz, l[[i]]];
    If[Total[parz] == Length[parz],
      AppendTo[res, parz];
      parz = {}
    ]
  ];
  Return[res]
]
```

So you can apply the above function to the sequences and you will get:

```
In[189]:= splitseq[s1]
Out[189]= {{1}, {-1, 3}, {-3, 5}, {-5, 1, -1, 9}, {-9, 11}, {-11, 1, -1, 15}, {-15, 17}, {-17, 1, -1, 21},
{-21, 3, -3, 1, -1, 27}, {-27, 29}, {-29, 1, -1, 3, -3, 35}, {-35, 1, -1, 39},
{-39, 41}, {-41, 1, -1, 45}, {-45, 5, -5, 1, -1, 51}, {-51, 3, -3, 1, -1, 57},
{-57, 59}, {-59, 1, -1, 3, -3, 65}, {-65, 1, -1, 69}, {-69, 71}, {-71, 1, -1, 5, -5, 77},
{-77, 1, -1, 81}, {-81, 3, -3, 1, -1, 87}, {-87, 5, -5, 1, -1, 3, -3, 95},
{-95, 1, -1, 99}, {-99, 101}, {-101, 1, -1, 105}, {-105, 107}, {-107, 1, -1, 111},
{-111, 3, -3, 1, -1, 5, -5, 9, -9, 1, -1, 3, -3, 125}, {-125, 1, -1, 129},
{-129, 5, -5, 1, -1, 135}, {-135, 137}, {-137, 1, -1, 9, -9, 3, -3, 1, -1, 147},
{-147, 149}, {-149, 1, -1, 3, -3, 155}, {-155, 1, -1, 5, -5, 161}, {-161, 1, -1, 165},
{-165, 11, -11, 1, -1, 171}, {-171, 3, -3, 1, -1, 177}, {-177, 179},
{-179, 1, -1, 3, -3, 9, -9, 1, -1, 189}, {-189, 191}, {-191, 1, -1, 195}, {-195, 197}}
```

```
In[190]:= splitseq[s2]
Out[190]= {{1}, {-1, 3}, {-2, 4}, {-5, 1, 2, 6}, {-8, 10}, {-6, -2, -3, 15}, {-14, 16}, {-14, 2, 4, 12},
{-20, 2, 8, -10, 4, 22}, {-24, 26}, {-29, 9, 6, -10, -4, 34}, {-18, -6, -8, 36}, {-34, 36},
{-32, -6, 18, 24}, {-44, 4, -2, 12, -4, 40}, {-50, 8, -4, 12, 10, 30}, {-54, 56},
{-30, -24, -5, 11, -2, 56}, {-50, 6, -16, 64}, {-68, 70}, {-36, -32, 14, -8, 2, 66},
{-74, -2, 38, 42}, {-76, 10, 26, -14, -18, 78}, {-84, 8, 10, 8, 16, -28, -16, 94},
{-90, 4, -6, 96}, {-84, 86}, {-90, -6, 46, 54}, {-104, 106}, {-98, 26, -30, 106},
{-94, 4, 6, -16, 46, -42, -12, 6, 50, -20, -10, -26, 2, 120}, {-125, 41, -30, 118},
{-120, 8, 48, -62, 12, 120}, {-114, 116}, {-132, 40, 24, -58, -10, 26, 44, -66, 30, 112},
{-144, 146}, {-132, -2, -6, 20, -18, 144}, {-78, -26, -48, 18, -20, 160},
{-122, -30, 72, 84}, {-160, 6, 4, 2, 24, 130}, {-144, -22, 4, 48, 30, 90}, {-174, 176},
{-168, 48, -38, 14, -6, -14, 30, -40, 12, 172}, {-188, 190}, {-96, -84, -6, 190}, {-186, 188}}
```

As you can see comparing above two sequences, **corresponding subsequences have same length**. But the astounding result is that if you couple each natural number starting with 2 with the corresponding item in the sequences **each prime corresponds with each first item of each subsequence**.

Main Results

*Each subsequence length corresponds to a gap between primes in both sequences, with *minrad* and with *maxrad*.

*With *minrad* each subsequence has other properties:

- +every first item of a subsequence is equal to last item of the preceding subsequence but with changed sign

- +excluding first and last item of a subsequence, other items are in couples $(n, -n)$

- +*minrad* values can be calculated easily for many values in the interval centered on a given number, so you can reconstruct the subsequence and hence the position of the prime numbers at the extremes of the interval