Proving Super Square Root Power Bounds on the Parity of the Partition Function:

(Numerical computations verifying some conjectured properties + suggestions on how to proceed with a formal proof of these results ...)

Author: *Maxie D. Schmidt (**maxieds@gmail.com**)*

Date: *Compiled on 10/3/2017 for Croot and Rolen*

Notes: *Please do not distribute of share these results. These computations are intended as a preview of the promised write up following the graduate exams this week.*

Section 1: Choosing the prime subsets to count (*note that choosing all primes q<= x leads to a negative estimate for N_e(x)*)

We / I conjecture that if we break up {1,2,...,x} into floor(x^{0.51}) subintervals, **I_p(t)** or **t^p < q prime < (t+1)^p**, for **p:=1.96 = 1/0.51** and choose the set
Q_x := {q_t : q is the last prime in the interval I_{1.96}(t) for 1 <= t <= floor(x^{0.51})},
that the following properties hold:
□ **|Q_x| = c * x^{0.51}** for all x sufficiently large

□ **I_{e,Q_x}(x) = #(duplicate indices q_1-G_j = q_2 - G_k for q1>q_2 in Q_x and k < j) ≈ 0** (for all cases of x numerically tested below / so far)
Thus we have |Q_x| primes satisfying the congruence identity === 0 (mod 2) proved from the identity involving the Mobius function, where we have by our counting argument that:
N_e(x) >= |Q_x| - I_{e,Q_x}(x) ≃ |Q_x| + o(x^{0.51})

The next table(s) compute certificates of the prime subsets q<= x (in the form of the **Q_x** defined above) that satisfy **|Q_x| = floor(x^{0.51})** and have **I_{e,Q_x}(x) = o(x^{0.51})** (in particular, **= 0** for all computed examples below).
These computations, *which I really should replicate in Sage / Python to compile and really check for these certificates when floor(x^{0.51}) is very large*, suggest that we have a good method for selecting the prime subsets Q_x which make our counting argument show that
N_e(x) >= c * x^{0.51} + o(x^{0.51}) for x sufficiently large ...

In[655]:=
```
Nex[x_] := Sum[If[Mod[PartitionsP[n], 2, 0] == 0, 1, 0], {n, 1, x}]
PrimeIn[lower_, upper_] :=
 Select[Table[Prime[m], {m, 1, upper}], # ≥ Ceiling[lower] && # ≤ Floor[upper] &]
PrimeIn[lower_, upper_, i_] := Select[Table[Prime[m], {m, 1, Floor[upper]}],
   # ≥ Ceiling[lower] && # ≤ Floor[upper] &][[i]]
s[t_, eps_] := PrimeIn[t^(2/(1+2*eps)), (t+1)^(2/(1+2*eps))]
GetPrimePairs[primes_] := Subsets[primes, {2}]
IndicatorFunction[PrimesSet_] :=
 Sum[If[qp[[1]] - qp[[2]] - (t^2 + (6 k + 1) t)/2 == 0, 1, 0] +
   If[qp[[1]] - qp[[2]] - (t^2 + (6 k - 1) t)/2 == 0, 1, 0] +
   If[qp[[1]] - qp[[2]] - (t^2 + (6 k + 1) t + 2 k)/2 == 0, 1, 0] +
   If[qp[[1]] - qp[[2]] - (t^2 + (6 k - 1) t - 2 k)/2 == 0, 1, 0],
  {qp, GetPrimePairs[PrimesSet]}, {t, 1, mu[qp[[1]]]}, {k, 1, t - 1}]
GetPrimeLists[x_, eps_] := Module[{},
   pints = Map[{Last[s[#1, eps]]} &, Range[1, Floor[Power[x, (1+2*eps)/2]]]];
   Return[Tuples[pints]];
  ];
Table[{Idx → x, Floor[Sqrt[x]], Floor[Power[x, 0.51]], Ne[x] → Nex[x], Certificate[x] ->
    Block[{s = Sort[Map[{IndicatorFunction[#1], #1} &, GetPrimeLists[x, 0.01]]]},
     Select[s, #[[1]] == s[[1]][[1]] &]]},
  {x, Table[Floor[Power[m, 1.96]], {m, 1, 75}]}] // TF
```

Out[662]//TableForm=

Idx → 1	1	1	Ne[1] → 0	Certificate[1] → {{0, {3}}}
Idx → 3	1	1	Ne[3] → 1	Certificate[3] → {{0, {3}}}
Idx → 8	2	2	Ne[8] → 2	Certificate[8] → {{0, {3, 7}}}
Idx → 15	3	3	Ne[15] → 6	Certificate[15] → {{0, {3, 7, 13}}}
Idx → 23	4	4	Ne[23] → 9	Certificate[23] → {{0, {3, 7, 13, 23}}}
Idx → 33	5	5	Ne[33] → 15	Certificate[33] → {{0, {3, 7, 13, 23, 31}}}
Idx → 45	6	6	Ne[45] → 19	Certificate[45] → {{0, {3, 7, 13, 23, 31, 43}}}
Idx → 58	7	7	Ne[58] → 25	Certificate[58] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 74	8	8	Ne[74] → 32	Certificate[74] → {{0, {3, 7, 13, 23, 31, 43, 5

Idx → 91	9	9	Ne[91] → 38	Certificate[91] → {{0, {3, 7, 13, 23, 31, 43, 53
Idx → 109	10	10	Ne[109] → 48	Certificate[109] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 130	11	11	Ne[130] → 61	Certificate[130] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 152	12	12	Ne[152] → 71	Certificate[152] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 176	13	13	Ne[176] → 82	Certificate[176] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 201	14	14	Ne[201] → 90	Certificate[201] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 229	15	15	Ne[229] → 98	Certificate[229] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 258	16	16	Ne[258] → 111	Certificate[258] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 288	16	17	Ne[288] → 123	Certificate[288] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 320	17	18	Ne[320] → 139	Certificate[320] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 354	18	19	Ne[354] → 158	Certificate[354] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 390	19	20	Ne[390] → 175	Certificate[390] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 427	20	21	Ne[427] → 194	Certificate[427] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 466	21	22	Ne[466] → 210	Certificate[466] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 507	22	23	Ne[507] → 227	Certificate[507] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 549	23	24	Ne[549] → 253	Certificate[549] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 593	24	25	Ne[593] → 275	Certificate[593] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 638	25	26	Ne[638] → 295	Certificate[638] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 686	26	27	Ne[686] → 320	Certificate[686] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 735	27	28	Ne[735] → 342	Certificate[735] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 785	28	29	Ne[785] → 370	Certificate[785] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 837	28	30	Ne[837] → 394	Certificate[837] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 891	29	31	Ne[891] → 417	Certificate[891] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 946	30	32	Ne[946] → 441	Certificate[946] → {{0, {3, 7, 13, 23, 31, 43, 5
Idx → 1003	31	33	Ne[1003] → 473	Certificate[1003] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1062	32	34	Ne[1062] → 508	Certificate[1062] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1122	33	35	Ne[1122] → 532	Certificate[1122] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1184	34	36	Ne[1184] → 562	Certificate[1184] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1248	35	37	Ne[1248] → 592	Certificate[1248] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1313	36	38	Ne[1313] → 632	Certificate[1313] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1380	37	39	Ne[1380] → 663	Certificate[1380] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1448	38	40	Ne[1448] → 699	Certificate[1448] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1519	38	41	Ne[1519] → 735	Certificate[1519] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1590	39	42	Ne[1590] → 774	Certificate[1590] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1664	40	43	Ne[1664] → 810	Certificate[1664] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1738	41	44	Ne[1738] → 849	Certificate[1738] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1815	42	45	Ne[1815] → 892	Certificate[1815] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1893	43	46	Ne[1893] → 929	Certificate[1893] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 1973	44	47	Ne[1973] → 975	Certificate[1973] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2054	45	48	Ne[2054] → 1010	Certificate[2054] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2137	46	49	Ne[2137] → 1056	Certificate[2137] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2222	47	50	Ne[2222] → 1100	Certificate[2222] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2308	48	51	Ne[2308] → 1139	Certificate[2308] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2396	48	52	Ne[2396] → 1182	Certificate[2396] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2485	49	53	Ne[2485] → 1226	Certificate[2485] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2576	50	54	Ne[2576] → 1268	Certificate[2576] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2669	51	55	Ne[2669] → 1320	Certificate[2669] → {{0, {3, 7, 13, 23, 31, 43,

Idx → 2763	52	56	Ne[2763] → 1369	Certificate[2763] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2859	53	57	Ne[2859] → 1419	Certificate[2859] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 2957	54	58	Ne[2957] → 1462	Certificate[2957] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3056	55	59	Ne[3056] → 1513	Certificate[3056] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3156	56	60	Ne[3156] → 1565	Certificate[3156] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3259	57	61	Ne[3259] → 1609	Certificate[3259] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3362	57	62	Ne[3362] → 1665	Certificate[3362] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3468	58	63	Ne[3468] → 1723	Certificate[3468] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3575	59	64	Ne[3575] → 1768	Certificate[3575] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3683	60	65	Ne[3683] → 1806	Certificate[3683] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3794	61	66	Ne[3794] → 1858	Certificate[3794] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 3905	62	67	Ne[3905] → 1905	Certificate[3905] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4019	63	68	Ne[4019] → 1962	Certificate[4019] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4134	64	69	Ne[4134] → 2024	Certificate[4134] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4250	65	70	Ne[4250] → 2089	Certificate[4250] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4368	66	71	Ne[4368] → 2157	Certificate[4368] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4488	66	72	Ne[4488] → 2214	Certificate[4488] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4609	67	73	Ne[4609] → 2275	Certificate[4609] → {{0, {3, 7, 13, 23, 31, 43,
Idx → 4732	68	74	Ne[4732] → 2340	Certificate[4732] → {{0, {3, 7, 13, 23, 31, 43,

Section 2: Proving that solutions leading to 1’s of our indicator function duplicate index counts satisfy certain particularly *“nice”* conjectured properties:

Note that we are looking at differences of primes of the form **q_t := ceiling(t^p) + d_t** where **q_t** is taken to be the last prime in the interval **I_p(t)** when **p:=1.96**.
Thus for **i>=1** and **t-i>=1**, we define **b := d_t-d_{t-i}**. Moreover, if t is sufficiently large and there is a prime at the ends of the intervals **I_p(t)** such that **d_{t+1} > floor((t+1)^p)-ceiling(t^p)**, then:

- **d_{t+1} > d_t => b >= 0** for all t
- **b != 0,1,2** for large enough t

Why are these properties noteworthy / important?
Well, see the computations below which suggest that any solutions (t, k, b) which make our defined ***IndicatorFunction*** > 0 appear to require that b ∈ {0,1,2} (or at least so far that b is considerably small relative to the respective interval sizes we are considering). So, it follows that if we can in fact ***prove*** that all solutions which contribute non-zero values of the indicator functions must have **b := 0,1,2** (*or say just sufficiently small relative to the primes we are taking differences of*) and we can show that since q_t is the *last prime* in its respective interval we have that **b>2** (*say*), then our total contribution of I_{e,Q_x}(x) = 0 (or is small and nicely bounded as o(x^{1/p}) when x is large).
In other words, these criteria being satisfied suffices to give a proof of our desired *super-square-root-power* bound!
(*See the following computations for evidence of this conjectured property of the b:*)

In[663]:=
```
Table[
  idx[s, i] -> Reduce[((Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k + 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k - 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] - (t^2 + (6 k + 1) t + 2 k) / 2 == 0 || Ceiling[s^p] -
          Ceiling[(s - i)^p] - (t^2 + (6 k - 1) t - 2 k) / 2 == 0) && 1 ≤ t ≤ Sqrt[s] &&
      1 ≤ k < t && 0 ≤ b ≤ Ceiling[(s + 1)^p - s^p] - Ceiling[(s + 1 - i)^p - (s - i)^p]) /.
    (p → 1.96), {b}, Integers], {s, 1, 50}, {i, 1, s - 1}] // TF
```

Out[663]//TableForm=

```
idx[2, 1] → False
idx[3, 1] → False                                                                                      idx[
idx[4, 1] → (k == 1 && t == 2 && b == 0) || (k == 1 && t == 2 && b == 2)                               idx[
idx[5, 1] → t == 2 && k == 1 && b == 1                                                                 idx[
idx[6, 1] → t == 2 && k == 1 && (b == 0 || b == 1)                                                     idx[
idx[7, 1] → False                                                                                      idx[
idx[8, 1] → False                                                                                      idx[
idx[9, 1] → (k == 1 && t == 3 && b == 0) || (k == 1 && t == 3 && b == 1) || (k == 1 && t == 3 && b == 2)   idx[
idx[10, 1] → False                                                                                     idx[
idx[11, 1] → False                                                                                     idx[
idx[12, 1] → t == 3 && k == 2 && b == 0                                                                idx[
idx[13, 1] → False                                                                                     idx[
idx[14, 1] → t == 3 && k == 2 && b == 0                                                                idx[
idx[15, 1] → False                                                                                     idx[
idx[16, 1] → (k == 2 && t == 4 && b == 0) || (k == 2 && t == 4 && b == 1)                             idx[
idx[17, 1] → t == 4 && k == 2 && b == 1                                                                idx[
idx[18, 1] → t == 4 && k == 2 && b == 0                                                                idx[
idx[19, 1] → False                                                                                     idx[
idx[20, 1] → t == 4 && k == 2 && b == 0                                                                idx[
idx[21, 1] → t == 4 && k == 2 && (b == 0 || b == 1 || b == 2)                                          idx[
idx[22, 1] → False                                                                                     idx[
idx[23, 1] → t == 4 && k == 3 && (b == 0 || b == 1 || b == 2)                                          idx[
idx[24, 1] → t == 4 && k == 3 && b == 1                                                                idx[
idx[25, 1] → k == 3 && t == 4 && b == 0                                                                idx[
idx[26, 1] → (t == 4 && k == 3 && b == 2) || (t == 5 && k == 2 && b == 1)                             idx[
idx[27, 1] → (t == 4 && k == 3 && b == 1) || (t == 5 && k == 2 && b == 0)                             idx[
idx[28, 1] → False                                                                                     idx[
idx[29, 1] → t == 4 && k == 3 && (b == 0 || b == 1 || b == 2)                                          idx[
idx[30, 1] → False                                                                                     idx[
idx[31, 1] → t == 5 && k == 3 && (b == 0 || b == 1)                                                    idx[
idx[32, 1] → t == 5 && k == 3 && b == 1                                                                idx[
idx[33, 1] → t == 5 && k == 3 && b == 0                                                                idx[
idx[34, 1] → False                                                                                     idx[
idx[35, 1] → t == 5 && k == 3 && b == 1                                                                idx[
idx[36, 1] → k == 3 && t == 5 && b == 0                                                                idx[
idx[37, 1] → False                                                                                     idx[
idx[38, 1] → False                                                                                     idx[
idx[39, 1] → False                                                                                     idx[
idx[40, 1] → t == 6 && k == 3 && b == 2                                                                idx[
idx[41, 1] → (t == 5 && k == 4 && b == 2) || (t == 6 && k == 3 && b == 1)                             idx[
idx[42, 1] → False                                                                                     idx[
idx[43, 1] → False                                                                                     idx[
idx[44, 1] → (t == 5 && k == 4 && b == 1) || (t == 6 && k == 3 && b == 1)                             idx[
idx[45, 1] → (t == 5 && k == 4 && b == 1) || (t == 6 && k == 3 && b == 1)                             idx[
idx[46, 1] → False                                                                                     idx[
idx[47, 1] → t == 6 && k == 3 && (b == 0 || b == 1)                                                    idx[
idx[48, 1] → False                                                                                     idx[
idx[49, 1] → (k == 3 && t == 7 && b == 0) || (k == 3 && t == 7 && b == 1)                             idx[
idx[50, 1] → (t == 6 && k == 4 && (b == 0 || b == 1 || b == 2)) || (t == 7 && k == 3 && b == 1)       idx[
```

In[670]:=
```
Table[
  idx[s, i] -> Reduce[((Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k + 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k - 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] - (t^2 + (6 k + 1) t + 2 k) / 2 == 0 || Ceiling[s^p] -
          Ceiling[(s - i)^p] - (t^2 + (6 k - 1) t - 2 k) / 2 == 0) && 1 ≤ t ≤ Sqrt[s] &&
      1 ≤ k < t && 0 ≤ b ≤ Ceiling[(s + 1)^p - s^p] - Ceiling[(s + 1 - i)^p - (s - i)^p]) /.
    (p → 1.96), {b}, Integers], {s, 51, 150}, {i, 1, s - 1}] // TF
```

On the other hand, if we instead chose q_t to be smaller than the last prime in the interval, say as in my first attempts which chose q_t to be the *first prime* in these intervals, then we have the possibility of negative values of b, which needless to say *significantly* complicates the formal calculations of the limiting behavior of **I_{e,Q_x}(x)**!
(*See, for example, below:*)

In[665]:=
```
Table[
  idx[s, i] -> Reduce[((Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k + 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] + b - (t^2 + (6 k - 1) t) / 2 == 0 ||
        Ceiling[s^p] - Ceiling[(s - i)^p] - (t^2 + (6 k + 1) t + 2 k) / 2 == 0 || Ceiling[s^p] -
          Ceiling[(s - i)^p] - (t^2 + (6 k - 1) t - 2 k) / 2 == 0) && 1 ≤ t ≤ Sqrt[s] &&
      1 ≤ k < t && - (Ceiling[(s + 1)^p - s^p] - Ceiling[(s + 1 - i)^p - (s - i)^p]) ≤
       b ≤ Ceiling[(s + 1)^p - s^p] - Ceiling[(s + 1 - i)^p - (s - i)^p]) /.
    (p → 1.96), {b}, Integers], {s, 1, 50}, {i, 1, s - 1}] // TF
```

Out[665]//TableForm=

```
idx[2, 1] → False
idx[3, 1] → False
idx[4, 1] → (k == 1 && t == 2 && b == 0) || (k == 1 && t == 2 && b == 2)
idx[5, 1] → t == 2 && k == 1 && (b == -1 || b == 1)
idx[6, 1] → t == 2 && k == 1 && (b == -1 || b == 0 || b == 1)
idx[7, 1] → False
idx[8, 1] → False
idx[9, 1] → (k == 1 && t == 3 && b == -2) || (k == 1 && t == 3 && b == -1) || (k == 1 && t == 3 && b == 0) || (k
idx[10, 1] → False
idx[11, 1] → False
idx[12, 1] → t == 3 && k == 2 && b == 0
idx[13, 1] → t == 3 && k == 2 && b == -1
idx[14, 1] → t == 3 && k == 2 && b == 0
idx[15, 1] → t == 3 && k == 2 && b == -1
idx[16, 1] → (k == 2 && t == 4 && b == -1) || (k == 2 && t == 4 && b == 0) || (k == 2 && t == 4 && b == 1)
idx[17, 1] → t == 4 && k == 2 && b == 1
idx[18, 1] → t == 4 && k == 2 && b == 0
idx[19, 1] → False
idx[20, 1] → t == 4 && k == 2 && b == 0
idx[21, 1] → t == 4 && k == 2 && (b == -2 || b == -1 || b == 0 || b == 1 || b == 2)
idx[22, 1] → False
idx[23, 1] → t == 4 && k == 3 && (b == -2 || b == -1 || b == 0 || b == 1 || b == 2)
idx[24, 1] → t == 4 && k == 3 && b == 1
idx[25, 1] → k == 3 && t == 4 && b == 0
idx[26, 1] → (t == 4 && k == 3 && (b == -2 || b == 2)) || (t == 5 && k == 2 && b == 1)
idx[27, 1] → (t == 4 && k == 3 && b == 1) || (t == 5 && k == 2 && b == 0)
idx[28, 1] → False
idx[29, 1] → t == 4 && k == 3 && (b == -2 || b == -1 || b == 0 || b == 1 || b == 2)
idx[30, 1] → False
idx[31, 1] → t == 5 && k == 3 && (b == -1 || b == 0 || b == 1)
idx[32, 1] → t == 5 && k == 3 && b == 1
idx[33, 1] → t == 5 && k == 3 && b == 0
idx[34, 1] → False
idx[35, 1] → t == 5 && k == 3 && b == 1
idx[36, 1] → k == 3 && t == 5 && b == 0
idx[37, 1] → t == 5 && k == 3 && b == -2
idx[38, 1] → False
idx[39, 1] → False
idx[40, 1] → t == 6 && k == 3 && b == 2
idx[41, 1] → (t == 5 && k == 4 && b == 2) || (t == 6 && k == 3 && b == 1)
idx[42, 1] → t == 5 && k == 4 && b == -1
idx[43, 1] → (t == 5 && k == 4 && b == -1) || (t == 6 && k == 3 && b == -2)
idx[44, 1] → (t == 5 && k == 4 && b == 1) || (t == 6 && k == 3 && b == 1)
idx[45, 1] → (t == 5 && k == 4 && b == 1) || (t == 6 && k == 3 && b == 1)
idx[46, 1] → (t == 5 && k == 4 && b == -2) || (t == 6 && k == 3 && b == -2)
idx[47, 1] → t == 6 && k == 3 && (b == -1 || b == 0 || b == 1)
idx[48, 1] → False
idx[49, 1] → (k == 3 && t == 7 && b == -1) || (k == 3 && t == 7 && b == 0) || (k == 3 && t == 7 && b == 1)
idx[50, 1] → (t == 6 && k == 4 && (b == -2 || b == -1 || b == 0 || b == 1 || b == 2)) || (t == 7 && k == 3 && b =
```

Section 3: The *"Analysis"* of this data ... Any notable suggestions?

Section 4: Misc relevant calculations to check identities:

In[113]:=
```
mu[q_] := Floor[(Sqrt[24 q - 23] + 1) / 6]
Gj[j_] := 1 / 2 Ceiling[j / 2] Ceiling[(3 j + 1) / 2]
```

In[666]:=
```
Table[G[j] → Gj[j], {j, 0, 21}]
Table[G[j] → j (3 j - 1) / 2, {j, 0, 12}]
Table[G[j] → j (3 j + 1) / 2, {j, 0, 12}]
```

Out[666]=
```
{G[0] → 0, G[1] → 1, G[2] → 2, G[3] → 5, G[4] → 7, G[5] → 12, G[6] → 15, G[7] → 22, G[8] → 26,
 G[9] → 35, G[10] → 40, G[11] → 51, G[12] → 57, G[13] → 70, G[14] → 77, G[15] → 92,
 G[16] → 100, G[17] → 117, G[18] → 126, G[19] → 145, G[20] → 155, G[21] → 176}
```

Out[667]=
```
{G[0] → 0, G[1] → 1, G[2] → 5, G[3] → 12, G[4] → 22, G[5] → 35, G[6] → 51,
 G[7] → 70, G[8] → 92, G[9] → 117, G[10] → 145, G[11] → 176, G[12] → 210}
```

Out[668]=
```
{G[0] → 0, G[1] → 2, G[2] → 7, G[3] → 15, G[4] → 26, G[5] → 40, G[6] → 57,
 G[7] → 77, G[8] → 100, G[9] → 126, G[10] → 155, G[11] → 187, G[12] → 222}
```

In[669]:=
```
Table[{Idx → q, MoebiusMu[q] + 1, 2 mu[q], PartitionsP[q - 1] +
    Sum[PartitionsP[q - 1 - Gj[k]] Power[-1, Ceiling[k / 2]], {k, 1, 2 mu[q]}],
   PartitionsP[q - 1] + Sum[PartitionsP[q - 1 - Gj[k]], {k, 1, 2 mu[q]}],
   Mod[PartitionsP[q - 1] + Sum[PartitionsP[q - 1 - Gj[k]], {k, 1, 2 mu[q]}], 2, 0]},
  {q, Table[Prime[m], {m, 1, 21}]}] // TF
```

Out[669]//TableForm=

Idx → 2	0	2	0	2	0
Idx → 3	0	2	0	4	0
Idx → 5	0	2	0	10	0
Idx → 7	0	4	0	24	0
Idx → 11	0	4	0	104	0
Idx → 13	0	6	0	198	0
Idx → 17	0	6	0	634	0
Idx → 19	0	6	0	1084	0
Idx → 23	0	8	0	2952	0
Idx → 29	0	8	0	11556	0
Idx → 31	0	8	0	17688	0
Idx → 37	0	10	0	59122	0
Idx → 41	0	10	0	125768	0
Idx → 43	0	10	0	181104	0
Idx → 47	0	10	0	367056	0
Idx → 53	0	12	0	1007036	0
Idx → 59	0	12	0	2622872	0
Idx → 61	0	12	0	3572064	0
Idx → 67	0	12	0	8779128	0
Idx → 71	0	14	0	15658128	0
Idx → 73	0	14	0	20791564	0