Making the table for $T = \frac{5}{7}, 1, 2.$

Validating the top of the table, by loading the zeros

In[1]:=
```
Andrew[5 / 7, 0, 0] = 42;
Andrew[5 / 7, 0, 1] = 172;
Andrew[5 / 7, 0, 2] = 934;
Andrew[5 / 7, 1, 0] = 16;
Andrew[5 / 7, 1, 1] = 66;
Andrew[5 / 7, 1, 2] = 934;

Andrew[1, 0, 0] = 36;
Andrew[1, 0, 1] = 148;
Andrew[1, 0, 2] = 844;
Andrew[1, 1, 0] = 12;
Andrew[1, 1, 1] = 42;
Andrew[1, 1, 2] = 408;
Andrew[1, 1, 3] = 844;

Andrew[2, 0, 0] = 16;
Andrew[2, 0, 1] = 28;
Andrew[2, 0, 2] = 120;
Andrew[2, 0, 3] = 330;
Andrew[2, 0, 4] = 634;
Andrew[2, 1, 0] = 10;
Andrew[2, 1, 1] = 18;
Andrew[2, 1, 2] = 64;
Andrew[2, 1, 3] = 210;
Andrew[2, 1, 4] = 630;
Andrew[2, 1, 5] = 634;

Andrew[T_, a_, k_] := Andrew[T, a, k - 1];
```

In[26]:=
```
<< DirichletCharacters`
```

In[27]:=
```
(*PC[q] is a list of primitive characters with conductor q,
including 1 from  each conjugate pair *)
(*PC[q,0] is a list of primitive characters with conductor q that are even,
including just 1 from each conjugate pair,
and PC[q,1] is the same for odd characters *)
PC[q_] := PC[q] = Select[PrimitiveCharacters[q],
    ConreyIndex[#] ≤ ConreyIndex[ConjugateCharacter[#]] &];
PC[q_, 0] := PC[q, 0] = Select[PC[q], EvenQ];
PC[q_, 1] := PC[q, 1] = Complement[PC[q], PC[q, 0]];
```

In[30]:=
```
AbsoluteTiming[Monitor[Do[PC[q]; PC[q, 0]; PC[q, 1], {q, 1, 934}], q]]
```

Out[30]=
```
{730.993, Null}
```

In[31]:=
```
(* This version of NTchi only works for characters in PC. *)
NTchi[T_, chi_] := Block[{max, zeros},
  zeros = CharacterData[chi][[Key["zeros"]]];
  Count[zeros, _? (Abs[#] ≤ T &)]]
```

The entries in the Table are correct

In[32]:=
```
(* The function Andrew[5/7,0,k] is defined correctly *)
(* T = 5/7 , 0≤k≤2 , a=0*)
Monitor[Block[{T = 5/7, a = 0},
  Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
     {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 2}]], q]
```

Out[32]=
```
{0, 1, 2}
```

In[33]:=
```
(* The function Andrew[5/7,1,k] is defined correctly  *)
(* T = 5/7 , 0≤k≤2 , a=1*)
Block[{T = 5 / 7, a = 1},
 Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
    {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 2}]]
```

Out[33]=
```
{0, 1, 2}
```

In[34]:=
```
(* The function Andrew[1,a,k] is defined correctly  *)
(* T = 1 *)
Block[{T = 1, a = 0},
 Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
    {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 2}]]

Block[{T = 1, a = 1},
 Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
    {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 3}]]
```

Out[34]= {0, 1, 2}

Out[35]= {0, 1, 2, 3}

In[36]:=
```
(* The function Andrew[2,a,k] is defined correctly  *)
(* T = 2 *)
Block[{T = 2, a = 0},
 Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
    {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 4}]]

Block[{T = 2, a = 1},
 Table[Max[Table[Max[Map[NTchi[T, #] &, PC[q, a]]],
    {q, If[k == 0, 1, Andrew[T, a, k - 1] + 1], Andrew[T, a, k]}]], {k, 0, 5}]]
```

Out[36]= {0, 1, 2, 3, 4}

Out[37]= {0, 1, 2, 3, 4, 5}

The number of characters and zeros being handled

In[38]:=
```
Sum[Length[PC[q]], {q, 3, 933}]
```

Out[38]= 80 818

In[39]:=
```
Monitor[Sum[Total[Map[Length[CharacterData[#][[Key["zeros"]]]] &, PC[q]]],
  {q, Select[Range[3, 933], Mod[#, 4] ≠ 2 &]}], q]
```

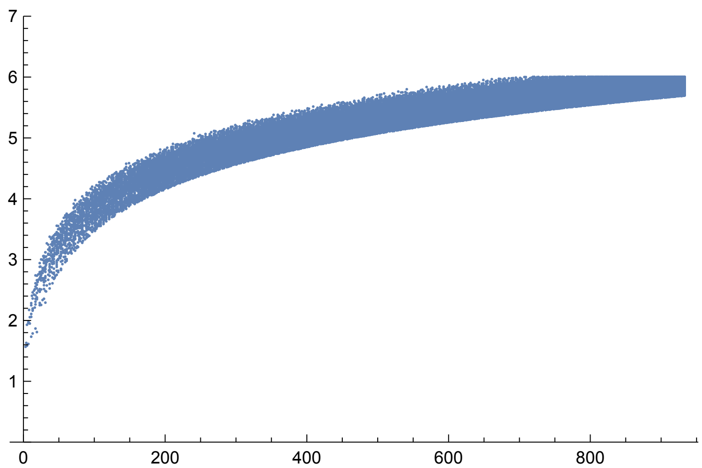
Out[39]= 403 272

N(T,χ)=0

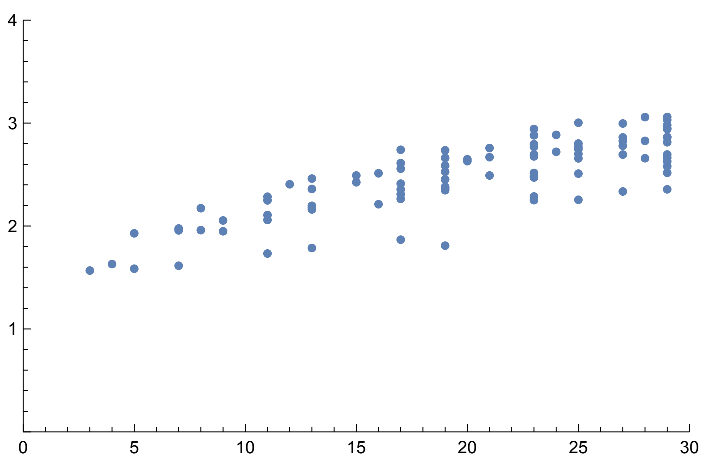
In[40]:=
```
Monitor[smallestells = Flatten[Table[(*for each q*)
      Table[(*for each chi*)
       max = CharacterData[chi][["maxheight"]];
       zeros = CharacterData[chi][["zeros"]];
       {q, Log[q (If[Length[zeros] == 0, max, Abs[zeros[[1]]] + 2) / (2 π)]},
       {chi, PC[q]}],
      {q, Select[Range[3, 934], Mod[#, 4] ≠ 2 &]}], 1];, q]
```

In[41]:=
```
ListPlot[smallestells, PlotRange → {0, 7}]
ListPlot[smallestells, PlotRange → {{0, 30}, {0, 4}}, PlotStyle → PointSize[Medium]]
```

Out[41]=



Out[42]=



The $\frac{\text{ell}}{\log(2+\text{ell})}$ bound

In[43]:=
```
flaws = {};
Monitor[Do[
   chars = PC[q, a];
   Do[(*for each character*)
    zeros = CharacterData[chi][["zeros"]];
    real = CharacterData[chi][["real"]];
    Do[(*for each zero*)
     {zl, zu} = Abs[zeros[[k]]] + 10^-12 {-2, 2};
     lterml = Log[q (zl+2)/(2 π)] / Log[2 + Log[q (zl+2)/(2 π)]];
     Tterml = zl/π Log[q zl/(2 π E)] + a/2 - 1/4;
     If[Not[If[real, 2, 1] * k ≤ Tterml + lterml], AppendTo[flaws, {chi, zeros[[k]]}]],
     {k, Length[zeros]}],
    {chi, chars}],
   {q, Select[Range[3, 934], Mod[#, 4] ≠ 2 &]}, {a, 0, 1}], {q, flaws}];
flaws
```

Out[45]= { }

In[46]:=
```
<< IntervalTools`
```

In[47]:=
```
flaws = {};
Do[
 chars = PC[q, a];
 Do[(*for each character*)
  zeros = CharacterData[chi][["zeros"]];
  real = CharacterData[chi][["real"]];
  Do[(*for each zero*)
   maxz = Abs[zeros[[k]]] + 2 × 10^-12;
   If[maxz ≥ 5/7 && maxz ≥ (2 (e^(197/125) π - q))/q,
    lower = Max[If[k == 1, 0, Abs[zeros[[k - 1]]] - 2/10^12], 5/7, (2 (e^(197/125) π - q))/q];
    upper = maxz;
    lterm[g_Interval] :=
     Interval[{Log[(q (Min[g]+2))/(2 π)]/Log[2 + Log[(q (Min[g]+2))/(2 π)]], Log[(q (Max[g]+2))/(2 π)]/Log[2 + Log[(q (Max[g]+2))/(2 π)]]}];
    Tterm[g_Interval] := If[Min[g] ≥ (2 π)/q || Max[g] ≤ (2 π)/q,
      Interval[{Min[g]/π Log[(q Min[g])/(2 π E)] + a/2 - 1/4, Max[g]/π Log[(q Max[g])/(2 π E)] + a/2 - 1/4}],
      Interval[a/2 - 1/4 + {0, Max[Min[g]/π Log[(q Min[g])/(2 π E)], Max[g]/π Log[(q Max[g])/(2 π E)]]}]];

    target = If[real, 2 k - 2, k - 1];
    F[T_] := target - (T/π Log[(q T)/(2 π E)] + a/2 - 1/4 - Log[(q (T+2))/(2 π)]/Log[2 + Log[(q (T+2))/(2 π)]]);
    gapF[g_Interval] := target - (Tterm[g] - lterm[g]);

    cert = ProveNonNegative[F, gapF, {Interval[{lower, upper}]}, MaxDepth → 10];

    If[Length[cert[[1]]] + Length[cert[[2]]] > 0, AppendTo[flaws, {chi, zeros[[k]]}]]],
   {k, Length[zeros]}],
  {chi, chars}],
 {q, Select[Range[3, 933], Mod[#, 4] ≠ 2 &]}, {a, 0, 1}]
```

In[49]:=
```
flaws
```

Out[49]= { }

Finding the best c, r.

We will use the stronger version of Backlund’s Trick, and only check afterwards that its hypotheses are satisfied.

The Gamma Spread: $E(a, d, T)$

In[50]:=
```
Er[a_, d_, T_] = (4 (4 + 3 π))/(45 ((17 + 2 a)^2 + 4 T^2)^(3/2)) - (4 T)/(3 ((17 + 2 a)^2 + 4 T^2)) +
  (8 + 6 π)/(45 ((17 + 2 a - 2 d)^2 + 4 T^2)^(3/2)) + (2 T)/(3 ((17 + 2 a - 2 d)^2 + 4 T^2)) + (8 + 6 π)/(45 ((17 + 2 a + 2 d)^2 + 4 T^2)^(3/2)) +
  (2 T)/(3 ((17 + 2 a + 2 d)^2 + 4 T^2)) + 2 ArcTan[(1 + 2 a)/(2 T)] + 2 ArcTan[(5 + 2 a)/(2 T)] + 2 ArcTan[(9 + 2 a)/(2 T)] +
  2 ArcTan[(13 + 2 a)/(2 T)] - 1/2 (15 + 2 a) ArcTan[(17 + 2 a)/(2 T)] - ArcTan[(1 + 2 a - 2 d)/(2 T)] -
  ArcTan[(5 + 2 a - 2 d)/(2 T)] - ArcTan[(9 + 2 a - 2 d)/(2 T)] - ArcTan[(13 + 2 a - 2 d)/(2 T)] +
  1/4 (15 + 2 a - 2 d) ArcTan[(17 + 2 a - 2 d)/(2 T)] - ArcTan[(1 + 2 a + 2 d)/(2 T)] - ArcTan[(5 + 2 a + 2 d)/(2 T)] -
  ArcTan[(9 + 2 a + 2 d)/(2 T)] - ArcTan[(13 + 2 a + 2 d)/(2 T)] + 1/4 (15 + 2 a + 2 d) ArcTan[(17 + 2 a + 2 d)/(2 T)] +
  1/2 T Log[1 + (17 + 2 a)^2/(4 T^2)] - 1/4 T Log[1 + (17 + 2 a - 2 d)^2/(4 T^2)] - 1/4 T Log[1 + (17 + 2 a + 2 d)^2/(4 T^2)];
```

Jensen Integral

In[51]:=
```
F[ell_, T_, c_, r_, theta_] := Module[{η, σ, t, bs},
    η = Min[18/(10 + 9 ell), 1/2];

    σ = c + r Cos[theta];
    t = r Sin[theta];

    Which[
     σ ≥ 1 + η,
        Log[Zeta[σ]],
     -η ≤ σ ≤ 1 + η,
        Log[Zeta[1 + η]] + (1 + η - σ)/2 ell + (1 + η - σ)/4 Log[((σ + 1)^2 + (Abs[t] + T)^2)/(T + 2)^2],
     σ < -η,
        bs = Round[σ];
        Log[Zeta[1 - σ]] + ((1 - 2 σ)/2) ell + ((1 - 2 σ + 2 bs)/4) Log[((σ - bs + 1)^2 + (Abs[t] + T)^2)/(T + 2)^2] +
      1/2 Sum[Log[((σ + k - 1)^2 + (Abs[t] + T)^2)/(T + 2)^2], {k, 1, -bs}]
    ](*end Which*)
  ];
```

In[52]:=
```
Theta[c_, r_, sigma_] :=
  Which[sigma ≥ c + r, 0, sigma ≤ c - r, π, True, ArcCos[(sigma - c)/r]];
```

```
In[53]:= JensenIntegral[ell_, T_, c_, r_, Method → "NumericalSimpleSplit"] :=
  Module[{η, p1, p2, p3, p4, howfar},
   η = Min[18/(10 + 9 ell), 1/2];

   p1 = NIntegrate[Hold[F[ell, T, c, r, theta]],
     {theta, 0, Theta[c, r, 1 + η]}, PrecisionGoal → 8, WorkingPrecision → 30];
   p2 = NIntegrate[Hold[F[ell, T, c, r, theta]], {theta, Theta[c, r, 1 + η],
      Theta[c, r, -η]}, PrecisionGoal → 8, WorkingPrecision → 30];
   p3 = NIntegrate[Hold[F[ell, T, c, r, theta]], {theta, Theta[c, r, -η],
      Theta[c, r, -1/2]}, PrecisionGoal → 8, WorkingPrecision → 30];
   howfar = -Round[c - r];
   p4 = Sum[NIntegrate[Hold[F[ell, T, c, r, theta]],
      {theta, Theta[c, r, -j + 1/2], Theta[c, r, -j - 1/2]},
      PrecisionGoal → 8, WorkingPrecision → 30], {j, 1, howfar}];

   p1 + p2 + p3 + p4];
```

```
In[54]:= JensenIntegral[ell_, T_, c_, r_, Method → "Rigorous"] :=
  Module[{η, Th, κ, σ, t, theta, L, p, howfar, zeta},
   (*Need  an interval version of Zeta;
   the special cases should never occur, but whatever *)
   zeta[g_Interval /; Min[g] ≥ 1] := Interval[
     {If[Max[g] == Infinity, 1, Zeta[Max[g]]], If[Min[g] == 1, ∞, Zeta[Min[g]]]}];

   η = Min[18/(10 + 9 ell), 1/2];
   Th[σ_] := Theta[c, r, σ];
   κ1 = (Th[-η] - Th[1 + η]) (1 + η - c)/2 -
     (π - Th[-η]) (c - 1/2) + r/2 (Sin[Th[-η]] + Sin[Th[1 + η]]);

   L[j_, theta_] = Log[((c + r Cos[theta]) + j)^2 + (Abs[r Sin[theta]] + T)^2)/(T + 2)^2];
   p[1] = κ1 * ell;
   p[2] = (Th[-η] - Th[1 + η]) Log[Zeta[1 + η]];

   p[3] =
    IntervalIntegrate[Log[zeta[c + r Cos[#]]] &, Interval[{0, Theta[c, r, 1 + η]}]];

   p[4] =
    IntervalIntegrate[Log[zeta[1 - (c + r Cos[#])]] &, Interval[{Theta[c, r, -η], π}]];

   p[5] = IntervalIntegrate[((1 + η - (c + r Cos[#]))/4 L[1, #]) &,
     Interval[{Theta[c, r, 1 + η], Theta[c, r, -η]}]];
   p[6] = IntervalIntegrate[((1 - 2 (c + r Cos[#]))/4 L[1, #]) &,
     Interval[{Theta[c, r, -η], Theta[c, r, -1/2]}]];

   howfar = -Round[c - r];

   p[7] = Sum[IntervalIntegrate[
      ((1 - 2 (c + r Cos[#]) - 2 j)/4 L[j + 1, #] + 1/2 Sum[L[k - 1, #], {k, 1, j}]) &,
      Interval[{Theta[c, r, -j + 1/2], Theta[c, r, -j - 1/2]}]], {j, 1, howfar}];

   Max[Sum[p[i], {i, 7}]]]
```

Assembled Bound

In[55]:=
```
NTchiUpperBound[a_, q_, T_, c_, r_, Method → method_String] :=
 Block[{σ1, δ, Eδ, Eσ, S, integral},
  σ1 = 1/2 + Sqrt[2] (c - 1/2);
  δ = 2 c - σ1 - 1/2;
  Eδ = Er[a, δ, T];
  Eσ = Er[a, σ1 - 1/2, T];
  (* σ1 > 1*)
  (* c > 1 *)
  (* r > (1+Sqrt[2]) (c-1/2) *)
  (* 1/4 ≤ δ ≤ 3/4 *)

  integral = JensenIntegral[Log[q (T + 2)/(2 π)], T, c, r, Method → method];

  S = Log[Zeta[c]/Zeta[2 c]] + 1/π integral;


  T/π Log[q/π] + 2/π Im[LogGamma[1/4 + a/2 + I * T/2]] +
   2/π Log[Zeta[σ1]] + 2/π (π S/(2 Log[r/(c-1/2)]) + Eδ/2 + (Eσ - Eδ)/2 (1 - Log[1 + Sqrt[2]]/Log[r/(c-1/2)]))]
```

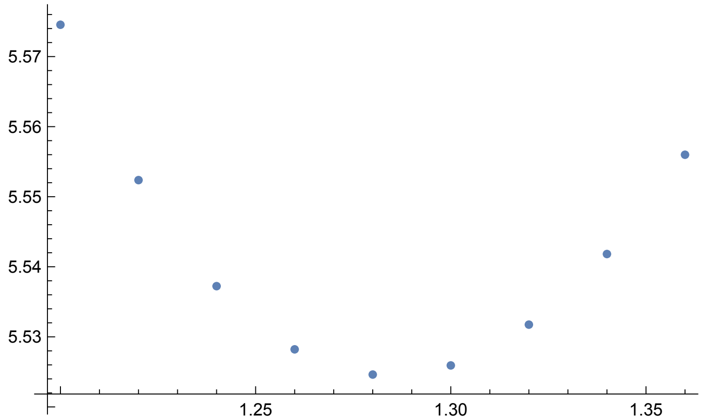
In[56]:=
```
AbsoluteTiming[NTchiUpperBound[0, 1000, 5 / 7, 1.21`, 1.9`, Method → "Rigorous"]]
AbsoluteTiming[
 NTchiUpperBound[0, 1000, 5 / 7, 1.21`, 1.9`, Method → "NumericalSimpleSplit"]]
```

Out[56]= {25.0749, 5.56257}

Out[57]= {0.052619, 5.56251}

In[58]:=
```
ListPlot[
 Table[{c, NTchiUpperBound[0, 1000, 5 / 7, c, 1.9`, Method → "NumericalSimpleSplit"]},
  {c, 1.2, 1.36, 0.02}]]
```

Out[58]=



Finding the Bounds

In[59]:=
```
Clear[Boundq];
```

In[60]:=
```
Boundq[a_Integer, k_, T_] := Boundq[a, k, T] =
   Module[{c = 1 + 1 / 8, r = 2 + 1 / 8, q, val, newval,
     done = False, newpairs, bests, newc, newr, newq, verbose = 0},

    {q, c, r} = Which[
      {a, k, T} == {0, 3, 5 / 7}, {172, 1.542`, 2.905`},
      {a, k, T} == {0, 4, 5 / 7}, {724, 1.446`, 2.642`},
      {a, k, T} == {0, 5, 5 / 7}, {3291, 1.378`, 2.444`},
      {a, k, T} == {0, 6, 5 / 7}, {15 978, 1.328, 2.292},
      {a, k, T} == {0, 7, 5 / 7}, {82 163, 1.289`, 2.171`},
      {a, k, T} == {0, 8, 5 / 7}, {443 025, 1.259`, 2.074`},
      {a, k, T} == {0, 9, 5 / 7}, {2 487 302, 1.234`, 1.993},

      {a, k, T} == {1, 3, 5 / 7}, {108, 1.727, 3.03},
      {a, k, T} == {1, 4, 5 / 7}, {435, 1.508, 2.741},
      {a, k, T} == {1, 5, 5 / 7}, {1908, 1.414, 2.527`},
      {a, k, T} == {1, 6, 5 / 7}, {9000, 1.352, 2.361},
      {a, k, T} == {1, 7, 5 / 7}, {45 099, 1.307, 2.229},
      {a, k, T} == {1, 8, 5 / 7}, {237 798, 1.273, 2.123},
      {a, k, T} == {1, 9, 5 / 7}, {1 309 289, 1.245, 2.035},

      {a, k, T} == {0, 3, 1}, {126, 1.584, 2.981},
      {a, k, T} == {0, 4, 1}, {438, 1.481, 2.736},
      {a, k, T} == {0, 5, 1}, {1615, 1.409, 2.545`},
      {a, k, T} == {0, 6, 1}, {6252, 1.356 , 2.393`},
```

```
{a, k, T} == {0, 7, 1}, {25 234, 1.316, 2.271`},
{a, k, T} == {0, 8, 1}, {105 519, 1.283, 2.17},
{a, k, T} == {0, 9, 1}, {454 852, 1.257, 2.086`},

{a, k, T} == {1, 3, 1}, {75, 1.701, 3.117},
{a, k, T} == {1, 4, 1}, {252, 1.539`, 2.842`},
{a, k, T} == {1, 5, 1}, {905, 1.446, 2.631`},
{a, k, T} == {1, 6, 1}, {3422, 1.382, 2.464`},
{a, k, T} == {1, 7, 1}, {13 544, 1.335, 2.33`},
{a, k, T} == {1, 8, 1}, {55 686, 1.299, 2.22},
{a, k, T} == {1, 9, 1}, {236 533, 1.27, 2.128`},
```

$\{a, k, T\} == \{0, 3, 2\}, \left\{46, \frac{213}{125}, \frac{3223}{1000}\right\},$

$\{a, k, T\} == \{0, 4, 2\}, \left\{110, \frac{791}{500}, \frac{3013}{1000}\right\},$

$\{a, k, T\} == \{0, 5, 2\}, \left\{266, \frac{1503}{1000}, \frac{2829}{1000}\right\},$

$\{a, k, T\} == \{0, 6, 2\}, \left\{660, \frac{1443}{1000}, \frac{669}{250}\right\},$

$\{a, k, T\} == \{0, 7, 2\}, \left\{1668, \frac{1397}{1000}, \frac{2549}{1000}\right\},$

$\{a, k, T\} == \{0, 8, 2\}, \left\{4287, \frac{34}{25}, \frac{1221}{500}\right\},$

$\{a, k, T\} == \{0, 9, 2\}, \left\{11\,179, \frac{133}{100}, \frac{47}{20}\right\},$

$\{a, k, T\} == \{1, 3, 2\}, \left\{30, \frac{9}{5}, \frac{419}{125}\right\},$

$\{a, k, T\} == \{1, 4, 2\}, \left\{71, \frac{327}{200}, \frac{1561}{500}\right\},$

$\{a, k, T\} == \{1, 5, 2\}, \left\{171, \frac{77}{50}, \frac{729}{250}\right\},$

$\{a, k, T\} == \{1, 6, 2\}, \left\{419, \frac{1471}{1000}, \frac{2749}{1000}\right\},$

$\{a, k, T\} == \{1, 7, 2\}, \left\{1049, \frac{1419}{1000}, \frac{261}{100}\right\},$

$\{a, k, T\} == \{1, 8, 2\}, \left\{2676, \frac{689}{500}, \frac{1247}{500}\right\},$

$\{a, k, T\} == \{1, 9, 2\}, \left\{6928, \frac{168}{125}, \frac{1197}{500}\right\},$

```
True, {724, 1.446`, 2.642`}

];
c = Round[c * 2^11] / 2^11;
```

```
r = Round[r * 2^11] / 2^11;

If[verbose > 1, Print["inside code"]];
val = NTchiUpperBound[a, q, T, c, r, Method -> "NumericalSimpleSplit"];
While[val > k + 1 - 2^-11,
 q = Round[0.95 q];
 val = NTchiUpperBound[a, q, T, c, r, Method -> "NumericalSimpleSplit"]];

If[verbose > 1, Print["Established  initial q of ", q]];
While[Not[done],

 (*climb q up as much as possible*)
 newq = Ceiling[1.01 q];
 newval = NTchiUpperBound[a, newq, T, c, r, Method -> "NumericalSimpleSplit"];
 While[newval ≤ k + 1 - 2^-11,
  If[verbose > 3, Print["EConsidering q of ", newq, " as newval = ", newval]];
  q = newq; val = newval; newq = Ceiling[1.01 q];
  newval =
   NTchiUpperBound[a, newq, T, c, r, Method -> "NumericalSimpleSplit"]];

 newq = q + 1;
 newval = NTchiUpperBound[a, newq, T, c, r, Method -> "NumericalSimpleSplit"];
 While[newval ≤ k + 1 - 2^-11,
  If[verbose > 3, Print["EConsidering q of ", newq, " as newval = ", newval]];
  q = newq;
  val = newval;
  newq = q + 1;
  newval = NTchiUpperBound[a, newq, T, c, r, Method -> "NumericalSimpleSplit"]];


 If[verbose > 1, Print["Bumped q to ", q, " with {c,r} of ", {c, r}]];

 (*so at this point  I have a reasonable q  *)
 newpairs =
  Select[{{-1/2048 + c, -1/2048 + r}, {-1/2048 + c, -1/128 + r}, {-1/2048 + c, -1/8 + r},
    {-1/2048 + c, r}, {-1/2048 + c, 1/8 + r}, {-1/2048 + c, 1/128 + r}, {-1/2048 + c, 1/2048 + r},
    {-1/128 + c, -1/2048 + r}, {-1/128 + c, -1/128 + r}, {-1/128 + c, -1/8 + r},
    {-1/128 + c, r}, {-1/128 + c, 1/8 + r}, {-1/128 + c, 1/128 + r}, {-1/128 + c, 1/2048 + r},
    {-1/8 + c, -1/2048 + r}, {-1/8 + c, -1/128 + r}, {-1/8 + c, -1/8 + r}, {-1/8 + c, r},
```

```
{-1/8 + c, 1/8 + r}, {-1/8 + c, 1/128 + r}, {-1/8 + c, 1/2048 + r}, {c, -1/2048 + r},
{c, -1/128 + r}, {c, -1/8 + r}, {c, r}, {c, 1/8 + r}, {c, 1/128 + r},
{c, 1/2048 + r}, {1/8 + c, -1/2048 + r}, {1/8 + c, -1/128 + r}, {1/8 + c, -1/8 + r},
{1/8 + c, r}, {1/8 + c, 1/8 + r}, {1/8 + c, 1/128 + r}, {1/8 + c, 1/2048 + r},
{1/128 + c, -1/2048 + r}, {1/128 + c, -1/128 + r}, {1/128 + c, -1/8 + r},
{1/128 + c, r}, {1/128 + c, 1/8 + r}, {1/128 + c, 1/128 + r}, {1/128 + c, 1/2048 + r},
{1/2048 + c, -1/2048 + r}, {1/2048 + c, -1/128 + r}, {1/2048 + c, -1/8 + r},
{1/2048 + c, r}, {1/2048 + c, 1/8 + r}, {1/2048 + c, 1/128 + r}, {1/2048 + c, 1/2048 + r}},
  First[#] > 1 && Last[#] > (1 + Sqrt[2]) (First[#] - 1/2) &];
bests = MinimalBy[newpairs, NTchiUpperBound[a, q, T, First[#],
    Last[#], Method -> "NumericalSimpleSplit"] &];
If[Length[bests] == 1, {newc, newr} = First[bests],
 Print["weirdness ", {c, r, newpairs, bests}];
 {newc, newr} = First[bests]];
If[{newc, newr} == {c, r}, done = True, {c, r, val} = {newc, newr,
   NTchiUpperBound[a, q, T, newc, newr, Method -> "NumericalSimpleSplit"]}]];

{q, c, r, val}];
```

First, $a = 0$, $T = 5/7$.

```
In[61]:= Boundq[0, 3, 5 / 7]
         Boundq[0, 4, 5 / 7]
         Boundq[0, 5, 5 / 7]
         Boundq[0, 6, 5 / 7]
         Boundq[0, 7, 5 / 7]
         Boundq[0, 8, 5 / 7]
         Boundq[0, 9, 5 / 7]
```

Out[61]= $\left\{172, \frac{3157}{2048}, \frac{5949}{2048}, 3.9954077343292828213785351279 5\right\}$

Out[62]= $\left\{724, \frac{2961}{2048}, \frac{5411}{2048}, 4.9993632514346258732057540491 8\right\}$

Out[63]= $\left\{3289, \frac{1411}{1024}, \frac{2503}{1024}, 5.9993305432633575507902658799 8\right\}$

Out[64]= $\left\{15\,991, \frac{2719}{2048}, \frac{2347}{1024}, 6.9994918462440447041782109904 9\right\}$

Out[65]= $\left\{82\,233, \frac{165}{128}, \frac{4447}{2048}, 7.9995083101829317129060952215 0\right\}$

Out[66]= $\left\{443\,412, \frac{2577}{2048}, \frac{2123}{1024}, 8.9995104759346714419289581062 8\right\}$

Out[67]= $\left\{2\,489\,523, \frac{2527}{2048}, \frac{4081}{2048}, 9.9995114931878261903755479687 9\right\}$

In[68]:=
```
Boundq[1, 3, 5 / 7]
Boundq[1, 4, 5 / 7]
Boundq[1, 5, 5 / 7]
Boundq[1, 6, 5 / 7]
Boundq[1, 7, 5 / 7]
Boundq[1, 8, 5 / 7]
Boundq[1, 9, 5 / 7]
```

Out[68]= $\left\{108, \frac{221}{128}, \frac{3103}{1024}, 3.992701145753053726010207950 15\right\}$

Out[69]= $\left\{435, \frac{3087}{2048}, \frac{5613}{2048}, 4.99859094451932137445599299091\right\}$

Out[70]= $\left\{1909, \frac{181}{128}, \frac{647}{256}, 5.99928890948222861961439098419\right\}$

Out[71]= $\left\{9007, \frac{1385}{1024}, \frac{1209}{512}, 6.99948845129756348518665927839\right\}$

Out[72]= $\left\{45137, \frac{2677}{2048}, \frac{2283}{1024}, 7.99950389380490023752851228358\right\}$

Out[73]= $\left\{238003, \frac{1303}{1024}, \frac{1087}{512}, 8.99950940624212345958751010104\right\}$

Out[74]= $\left\{1310445, \frac{1275}{1024}, \frac{521}{256}, 9.99951146458858053325718190564\right\}$

In[75]:=
```
Boundq[0, 3, 1]
Boundq[0, 4, 1]
Boundq[0, 5, 1]
Boundq[0, 6, 1]
Boundq[0, 7, 1]
Boundq[0, 8, 1]
Boundq[0, 9, 1]
```

Out[75]= $\left\{126, \frac{3245}{2048}, \frac{3053}{1024}, 3.99618014119430749282017888748\right\}$

Out[76]= $\left\{438, \frac{3033}{2048}, \frac{1401}{512}, 4.99800340166729758668354637427\right\}$

Out[77]= $\left\{1616, \frac{1443}{1024}, \frac{1303}{512}, 5.99938722101377827948080848532\right\}$

Out[78]= $\left\{6256, \frac{1389}{1024}, \frac{2451}{1024}, 6.99944733474058428262973156981\right\}$

Out[79]= $\left\{25\,252, \frac{1347}{1024}, \frac{4651}{2048}, 7.99950339034654404738697229254\right\}$

Out[80]= $\left\{105\,597, \frac{657}{512}, \frac{1111}{512}, 8.99950583684477760257988235471\right\}$

Out[81]= $\left\{455\,195, \frac{2575}{2048}, \frac{267}{128}, 9.99951084345254370547263556404\right\}$

```
In[82]:= Boundq[1, 3, 1]
Boundq[1, 4, 1]
Boundq[1, 5, 1]
Boundq[1, 6, 1]
Boundq[1, 7, 1]
Boundq[1, 8, 1]
Boundq[1, 9, 1]
```

Out[82]= $\left\{75, \frac{3485}{2048}, \frac{399}{128}, 3.994027458372605482833085199255\right\}$

Out[83]= $\left\{252, \frac{3151}{2048}, \frac{5821}{2048}, 4.996876073809124583353935921088\right\}$

Out[84]= $\left\{905, \frac{2961}{2048}, \frac{1347}{512}, 5.998987058673452006307798951988\right\}$

Out[85]= $\left\{3425, \frac{2831}{2048}, \frac{2523}{1024}, 6.999470782993147751610389391155\right\}$

Out[86]= $\left\{13554, \frac{1367}{1024}, \frac{4771}{2048}, 7.999489048634154340779801449533\right\}$

Out[87]= $\left\{55727, \frac{665}{512}, \frac{2273}{1024}, 8.999506971937795636066464150299\right\}$

Out[88]= $\left\{236710, \frac{325}{256}, \frac{2179}{1024}, 9.999509192352631085039705389166\right\}$

In[89]:=
```
Boundq[0, 3, 2]
Boundq[0, 4, 2]
Boundq[0, 5, 2]
Boundq[0, 6, 2]
Boundq[0, 7, 2]
Boundq[0, 8, 2]
Boundq[0, 9, 2]
```

Out[89]= $\left\{46, \frac{1745}{1024}, \frac{6601}{2048}, 3.9807652751148867586679372187 4\right\}$

Out[90]= $\left\{110, \frac{405}{256}, \frac{3085}{1024}, 4.99840522982586708124184672220\right\}$

Out[91]= $\left\{266, \frac{1539}{1024}, \frac{5793}{2048}, 5.99692102515711424920632606293\right\}$

Out[92]= $\left\{660, \frac{739}{512}, \frac{5481}{2048}, 6.99841059450475385987117457216\right\}$

Out[93]= $\left\{1669, \frac{2861}{2048}, \frac{5221}{2048}, 7.99923743653549333158591729171\right\}$

Out[94]= $\left\{4289, \frac{2785}{2048}, \frac{5001}{2048}, 8.99930176627109595910194643552\right\}$

Out[95]= $\left\{11185, \frac{2723}{2048}, \frac{1203}{512}, 9.99950269724653407671154780649\right\}$

In[96]:=
```
Boundq[1, 3, 2]
Boundq[1, 4, 2]
Boundq[1, 5, 2]
Boundq[1, 6, 2]
Boundq[1, 7, 2]
Boundq[1, 8, 2]
Boundq[1, 9, 2]
```

Out[96]= $\left\{30, \frac{3687}{2048}, \frac{6865}{2048}, 3.964321169327705459802869990 40\right\}$

Out[97]= $\left\{71, \frac{837}{512}, \frac{3197}{1024}, 4.9889356097147818569279642961 3\right\}$

Out[98]= $\left\{171, \frac{3153}{2048}, \frac{5971}{2048}, 5.9980577775427199842451849903 7\right\}$

Out[99]= $\left\{419, \frac{3013}{2048}, \frac{5629}{2048}, 6.9978861012522562861667920627 7\right\}$

Out[100]= $\left\{1050, \frac{1453}{1024}, \frac{2673}{1024}, 7.9990847752840016686881343689 4\right\}$

Out[101]= $\left\{2677, \frac{1411}{1024}, \frac{5107}{2048}, 8.9993274345082508554023532416 2\right\}$

Out[102]= $\left\{6932, \frac{2753}{2048}, \frac{613}{256}, 9.9994940057011650231878688248 3\right\}$

Validating the bottom of the table

In[103]:=
```
AbsoluteTiming[Block[{a = 0, T = 5 / 7, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[103]= {1116.02, {True, True, True, True, True, True, True}}

In[104]:=
```
AbsoluteTiming[Block[{a = 1, T = 5 / 7, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[104]= {1142.39, {True, True, True, True, True, True, True}}

In[105]:=
```
AbsoluteTiming[Block[{a = 0, T = 1, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[105]= {1158.23, {True, True, True, True, True, True, True}}

In[106]:=
```
AbsoluteTiming[Block[{a = 1, T = 1, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[106]=
```
{1152.68, {True, True, True, True, True, True, True}}
```

In[107]:=
```
AbsoluteTiming[Block[{a = 0, T = 2, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[107]=
```
{1022.68, {True, True, True, True, True, True, True}}
```

In[108]:=
```
AbsoluteTiming[Block[{a = 1, T = 2, q, c, r, int, k},
  Table[
   {q, c, r, int} = Boundq[a, k, T];
   int < NTchiUpperBound[a, q, T, c, r, Method → "Rigorous"] < k + 1, {k, 3, 9}]]]
```

Out[108]=
```
{1214.48, {True, True, True, True, True, True, True}}
```

Writing the Tables

Kevin’s Tables

In[109]:=
```
Kevin[T_, a_, k_ /; 3 ≤ k ≤ 9] := Boundq[a, k, T][[1]];
Kevin[T_, a_, k_] := 1;
```

In[111]:=
```
TableForm[Table[{k, Kevin[5 / 7, 0, k], Kevin[5 / 7, 1, k],
   Kevin[1, 0, k], Kevin[1, 1, k], Kevin[2, 0, k], Kevin[2, 1, k]}, {k, 0, 9}]]
```

Out[111]//TableForm=

0	1	1	1	1	1	1
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	172	108	126	75	46	30
4	724	435	438	252	110	71
5	3289	1909	1616	905	266	171
6	15 991	9007	6256	3425	660	419
7	82 233	45 137	25 252	13 554	1669	1050
8	443 412	238 003	105 597	55 727	4289	2677
9	2 489 523	1 310 445	455 195	236 710	11 185	6932

In[112]:=
```
TableForm[{Block[{a = 0, T = 5 / 7},
   Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]], Block[
   {a = 1, T = 5 / 7}, Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]],
  Block[{a = 0, T = 1}, Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]],
  Block[{a = 1, T = 1}, Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]],
  Block[{a = 0, T = 2}, Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]],
  Block[{a = 1, T = 2}, Table[{Boundq[a, k, T][[2]], Boundq[a, k, T][[3]]} 2^11, {k, 5, 9}]]}]
```

Out[112]//TableForm=

2822 5006	2719 4694	2640 4447	2577 4246	2527 4081
2896 5176	2770 4836	2677 4566	2606 4348	2550 4168
2886 5212	2778 4902	2694 4651	2628 4444	2575 4272
2961 5388	2831 5046	2734 4771	2660 4546	2600 4358
3078 5793	2956 5481	2861 5221	2785 5001	2723 4812
3153 5971	3013 5629	2906 5346	2822 5107	2753 4904

Andrew’s Table

In[113]:=
```
TableForm[Table[{k, Andrew[5 / 7, 0, k], Andrew[5 / 7, 1, k], Andrew[1, 0, k],
   Andrew[1, 1, k], Andrew[2, 0, k], Andrew[2, 1, k]}, {k, 0, 9}]]
```

Out[113]//TableForm=

0	42	16	36	12	16	10
1	172	66	148	42	28	18
2	934	934	844	408	120	64
3	934	934	844	844	330	210
4	934	934	844	844	634	630
5	934	934	844	844	634	634
6	934	934	844	844	634	634
7	934	934	844	844	634	634
8	934	934	844	844	634	634
9	934	934	844	844	634	634

Combined Table

In[114]:=
```
TableForm[Table[{k, Max[Kevin[5 / 7, 0, k], Andrew[5 / 7, 0, k]],
   Max[Kevin[5 / 7, 1, k], Andrew[5 / 7, 1, k]], Max[Kevin[1, 0, k], Andrew[1, 0, k]],
   Max[Kevin[1, 1, k], Andrew[1, 1, k]], Max[Kevin[2, 0, k], Andrew[2, 0, k]],
   Max[Kevin[2, 1, k], Andrew[2, 1, k]]}, {k, 0, 9}]]
```

Out[114]//TableForm=

0	42	16	36	12	16	10
1	172	66	148	42	28	18
2	934	934	844	408	120	64
3	934	934	844	844	330	210
4	934	934	844	844	634	630
5	3289	1909	1616	905	634	634
6	15 991	9007	6256	3425	660	634
7	82 233	45 137	25 252	13 554	1669	1050
8	443 412	238 003	105 597	55 727	4289	2677
9	2 489 523	1 310 445	455 195	236 710	11 185	6932

In[115]:=
```
TeXForm[%]
```

Out[115]//TeXForm=

```
\left(
\begin{array}{ccccccc}
 0 & 42 & 16 & 36 & 12 & 16 & 10 \\
 1 & 172 & 66 & 148 & 42 & 28 & 18 \\
 2 & 934 & 934 & 844 & 408 & 120 & 64 \\
 3 & 934 & 934 & 844 & 844 & 330 & 210 \\
 4 & 934 & 934 & 844 & 844 & 634 & 630 \\
 5 & 3289 & 1909 & 1616 & 905 & 634 & 634 \\
 6 & 15991 & 9007 & 6256 & 3425 & 660 & 634 \\
 7 & 82233 & 45137 & 25252 & 13554 & 1669 & 1050 \\
 8 & 443412 & 238003 & 105597 & 55727 & 4289 & 2677 \\
 9 & 2489523 & 1310445 & 455195 & 236710 & 11185 & 6932 \\
\end{array}
\right)
```