

Some remarks on the usage of this notebook:

- The package HolonomicFunctions can be downloaded from the website <http://www.risc.uni-linz.ac.at/research/combinat/software/HolonomicFunctions/>
- Throughout this notebook, the variable qn is identified with the expression q^n (and analogously for qi and qj).
- The q -shift which sends qn to q^*qn and q^n to q^{n+1} is denoted by $QS[qn, q^n]$.

The objective of this notebook is to give a fully rigorous computer proof of the long-standing q -TSPP conjecture, which has been reduced to a determinant evaluation by Soichi Okada, which can be further reduced by the holonomic ansatz to proving three q -holonomic identities:

$$\text{Normalization: } c(n, n) = 1 \quad (n \geq 1)$$

$$\text{Soichi: } \sum_{j=1}^n a(i, j) c(n, j) = 0 \quad (1 \leq i < n)$$

$$\text{Okada: } \sum_{j=1}^n a(n, j) c(n, j) = \frac{b(n)}{b(n-1)} \quad (n \geq 1)$$

$$\text{where } a(i, j) = (q^i + 1) \delta_{i,j} - \delta_{i,j+1} + q^{i+j-1} \left(\binom{i+j-2}{i-1}_q + q \binom{i+j-1}{i}_q \right),$$

$$\text{and } b(n) = \prod_{i=1}^n \prod_{j=i}^n \prod_{k=j}^n \left(\frac{1-q^{i+j+k-1}}{1-q^{i+j+k-2}} \right)^2,$$

and $c(n, j)$ is another discrete function to be defined below.

```
<< "HolonomicFunctions.m"
```

HolonomicFunctions package by Christoph Koutschan, RISC-Linz, Version 1.3 (25.01.2010)
→ Type ?HolonomicFunctions for help

```
(* define the entries of the matrix *)
Clear[a];
a[i_Integer, j_Integer] := a[i, j] =
  Expand[
    FunctionExpand[q^(i+j-1) * (QBinomial[i+j-2, i-1, q] + q * QBinomial[i+j-1, i, q]) +
      (1+q^i) * KroneckerDelta[i, j] - KroneckerDelta[i, j+1]]];
```

For convenience, we define a simple function that gives all non-negative integer roots of a "univariate" q -polynomial. By that we mean a polynomial in q and some other variable that represents a power of q , e.g., q^n . The output then is a list of values $n \geq 0$ such that the given polynomial is zero.

```
In[22]:= MyQRoots[poly_, var_] :=
  With[{factors = First /@ FactorList[poly]},
    If[MemberQ[factors, 0],
      "Polynomial is identically zero.",
      Sort[Exponent[#, q] & /@
        Select[factors, MatchQ[#, (var - q^_.) | (q^_. - var) | (1 - var) | (var - 1)] &]]]
```

Our definition of $c(n, j)$

We pull out of the hat the following annihilating ideal, and use it to define a bivariate sequence uniquely.

```
annc = << "ann-qTSPP-deg.m";
```

The question which initial values have to be given in order to define the sequence uniquely, is considered in the following. In particular we do a detailed analysis of the singularities. For this purpose we extract the leading coefficients of the Gröbner basis `annc` and shift them according to the leading power products (such that if one of these polynomials is zero for some specific (n_0, j_0) then the value of $c(n_0, j_0)$ cannot be computed by means of the corresponding recurrence).

```
lcfs = (LeadingCoefficient[#, /. Thread[{qn, qj} → {qn, qj} / q^LeadingExponent[#]]) & /@ annc;
```

We want to define $c(n, j)$ for $n \geq 1$ and $j \geq 0$. The latter is just for convenience: when we are going to prove (Soichi) it will simplify things if we know that our definition implies $c(n, 0) = 0$ for all n . (to be shown later).

With these restrictions it is clear that in the area $(4, 5) + \mathbb{N}^2$ (i.e., all points with $j \geq 4$ and $n \geq 5$) all recurrences may be used to compute values of $c(n, j)$ without involving values from outside our definition range. The remaining points, two strips of width 4 along the coordinate axes, are investigated separately.

■ The strip $1 \leq n \leq 4$

`LeadingPowerProduct /@ annc`

$$\{S_{qj,q}^4, S_{qn,q} S_{qj,q}^3, S_{qn,q}^2 S_{qj,q}^2, S_{qn,q}^3 S_{qj,q}, S_{qn,q}^4\}$$

It is clear that in the case $n=1$, we can only use the first recurrence, for $n=2$ either the first or the second one, and so on.

$n = 1$:

```
MyQRroots[# /. qn -> q, qj] & /@ Take[lcfs, 1]
{{0, 1, 2, 3, 4, 5, 6}}

singn = {1, #} & /@ First[%]
{{1, 0}, {1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}}
```

$n = 2$:

```
MyQRroots[# /. qn -> q^2, qj] & /@ Take[lcfs, 2]
{{2, 3, 4}, {1, 2, 3, 4, 5, 7}}

singn = Join[singn, {{2, 0}, {2, 1}, {2, 2}, {2, 3}, {2, 4}}];
```

$n = 3$:

```
MyQRroots[# /. qn -> q^3, qj] & /@ Take[lcfs, 3]
{{3, 4}, {2, 3, 8}, {0, 1, 2, 3, 4}}

singn = Join[singn, {{3, 0}, {3, 1}, {3, 2}, {3, 3}}];
```

$n = 4$:

```
MyQRroots[# /. qn -> q^4, qj] & /@ Take[lcfs, 4]
{{3, 4, 5}, {2, 3, 4, 9}, {1, 2}, {0, 1, 2, 3, 5}}

singn = Join[singn, {{4, 0}, {4, 1}, {4, 2}}];
```

■ The strip $0 \leq j \leq 3$

$j = 0$:

```
MyQRroots[# /. qj -> 1, qn] & /@ Take[lcfs, -1]
{{0, 1, 2, 3, 4}}
```

```

singj = {#, 0} & /@Rest[First[%]]
{{1, 0}, {2, 0}, {3, 0}, {4, 0}}

```

j = 1:

```

MyQRroots[# /. qj → q, qn] & /@Take[lcfs, -2]
{Polynomial is identically zero., {0, 1, 2, 3, 4, 5}}
singj = Join[singj, {{1, 1}, {2, 1}, {3, 1}, {4, 1}, {5, 1}}];

```

j = 2:

```

MyQRroots[# /. qj → q^2, qn] & /@Take[lcfs, -3]
{Polynomial is identically zero., {1, 2, 3, 4}, {1, 2, 3, 4, 5}}
singj = Join[singj, {{1, 2}, {2, 2}, {3, 2}, {4, 2}}];

```

j = 3:

```

MyQRroots[# /. qj → q^3, qn] & /@Take[lcfs, -4]
{Polynomial is identically zero., {2, 3}, {1, 2, 3, 4}, {1, 2, 3, 4, 5}}
singj = Join[singj, {{1, 3}, {2, 3}, {3, 3}}];

```

■ the area $(4, 5) + \mathbb{N}^2$

We show that the leading coefficient of the third recurrence does not have any integer zeros in the area under consideration. A fortiori there cannot be any common zeros.

```

lcf = lcfs[[3]];

```

A numerical test before we start to be rigorous:

```

test = lcf /. q → 2;
Position[Table[test /. qn → 2^n /. qj → 2^j, {j, 4, 20}, {n, 5, 20}], 0, {2}]
{}

```

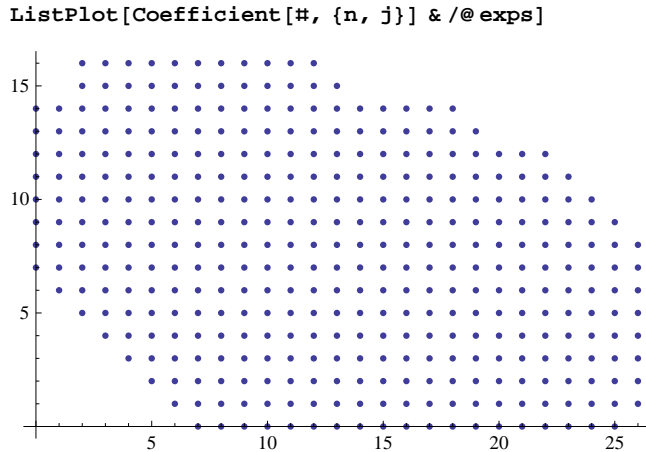
First we are extracting all the exponents that appear in the expanded leading coefficient. For each choice of n and j, we pick only the one with the highest power in q.

```

exps = DeleteCases[Flatten[MapIndexed[
  (Exponent[#1, q] + {n, j} . (#2 - 1)) &, CoefficientList[fac, {qn, qj}], {2}]], -Infinity];

```

In this picture we see all exponents of q^n (horizontally) and q^j (vertically).



Now we sort out all exponents that are always smaller than some other (again, inside the area under consideration). What we get is the upper right border of the above figure (this was somehow to be expected).

```

i1 = 1;
While[i1 ≤ Length[exps],
  i2 = 1;
  While[i2 ≤ Length[exps] && Simplify[exps[[i1]] < exps[[i2]], n ≥ 5 && j ≥ 4] != True, i2++];
  If[i2 > Length[exps], i1++, exps = Drop[exps, {i1}]];
]

```

```

exps

{31 + 16 j + 12 n, 33 + 15 j + 13 n, 32 + 14 j + 18 n, 34 + 13 j + 19 n,
 34 + 12 j + 22 n, 32 + 11 j + 23 n, 36 + 10 j + 24 n, 34 + 9 j + 25 n, 31 + 8 j + 26 n}

```

Now we compute conditions on n and j , under which each of the above exponents becomes the maximal one.

```

Table[
  {k, " ", Simplify[Reduce[Simplify[n ≥ 5 && j ≥ 4 && And @@ (exps[[k]] > # & /@ Drop[exps, {k}])]],
    Element[{n, j}, Integers]}], {k, Length[exps]}] // TableForm

```

1	$n \geq 5 \ \&\& \ j \geq 1 + 3 n$
2	False
3	$n \geq 5 \ \&\& \ j \geq 2 + 2 n \ \&\& \ j \leq 3 n$
4	False
5	$n \geq 5 \ \&\& \ j \geq 2 + n \ \&\& \ j \leq 2 n$
6	False
7	$j \leq n \ \&\& \ ((n = 5 \ \&\& \ j \geq 4) \ \ (n \geq 6 \ \&\& \ n \leq 1 + j))$
8	False
9	$n > 7 \ \&\& \ j \geq 4 \ \&\& \ 3 + j < n$

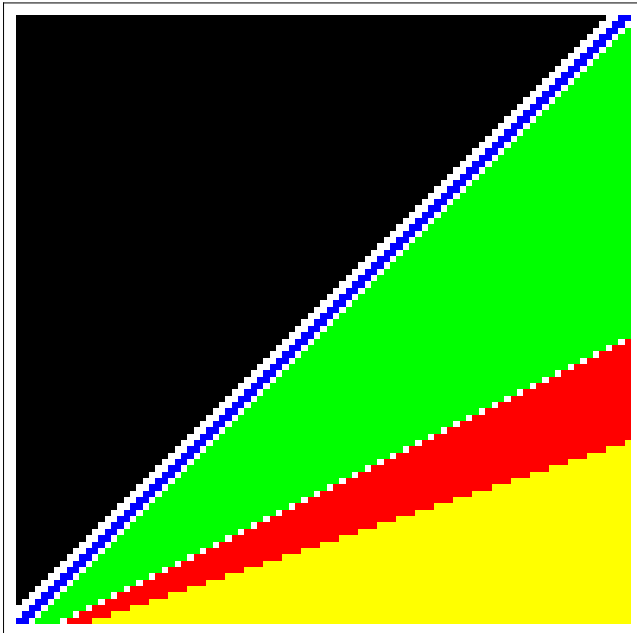
The entry at position 7 simplifies to $n \geq 5 \ \&\& \ n-1 \leq j \leq n$.

So we see that all points are covered, except the lines $j = 2n+1$, $j = n+1$, $j = n-2$, and $j = n-3$, as it can also be seen in the following picture:

```

ArrayPlot[Table[With[{pos = Position[exps, Max[exps]]}, If[Length[pos] == 1, pos[[1, 1]], 0]],
  {n, 100, 5, -1}, {j, 4, 100}],
ColorRules -> {0 -> White, 1 -> Yellow, 3 -> Red, 5 -> Green, 7 -> Blue, 9 -> Black}]

```



For these exceptional lines, we test separately whether the leading coefficient becomes zero (and find that none of the zeros lies in the area under consideration).

```
MyQRroots[lcf /. qj -> qn^2 * q, qn]
```

```
{0}
```

```
MyQRroots[lcf /. qj -> qn * q, qn]
```

```
{0, 1, 2, 3}
```

```
MyQRroots[lcf /. qn -> qj * q^2, qj]
```

```
{0, 1, 2}
```

```
MyQRroots[lcf /. qn -> qj * q^3, qj]
```

```
{0, 1, 2}
```

■ Conclusion

We have seen that there are no singularities for $n \geq 5$ and $j \geq 4$.

Therefore the only singularities are the ones that we have found in the strips $1 \leq n \leq 4$ and $0 \leq j \leq 3$.

```
sing = Union[singn, singj]
```

```
{1, 0}, {1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {2, 0}, {2, 1}, {2, 2},
{2, 3}, {2, 4}, {3, 0}, {3, 1}, {3, 2}, {3, 3}, {4, 0}, {4, 1}, {4, 2}, {5, 1}
```

We can hence write down a procedure that computes the value of $c(n, j)$ for given integers n and j , using our annihilating ideal plus the above initial values (although this is not the most efficient way).

```

Clear[c];
localAnnc = OrePolynomialSubstitute[annc, {qn → q^n, qj → q^j}] /. {n → ln, j → lj};
c[nn_Integer /; nn ≥ 1, jj_Integer /; jj ≥ 0] := c[nn, jj] =
  Which[
    (* special initial value (j=n=1) *)
    jj === nn === 1, 1,
    (* special initial values (j=0) *)
    jj === 0 && 1 ≤ nn ≤ 4, 0,
    (* special initial values (j>n) *)
    MemberQ[{{1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {2, 3}, {2, 4}}, {nn, jj}], 0,
    (* initial values that need to be computed *)
    MemberQ[{{2, 1}, {2, 2}, {3, 1}, {3, 2}, {3, 3}, {4, 1}, {4, 2}, {5, 1}}, {nn, jj}],
    Together[(-1)^(nn + jj) * Det[Table[Drop[Table[a[i1, j1], {j1, nn}], {jj}], {i1, nn - 1}]] /
      Det[Table[a[i1, j1], {i1, nn - 1}, {j1, nn - 1}]]],
    (* use recurrences of annc *)
    True,
    Together[Solve[
      Together[0 == ApplyOreOperator[#, c1[ln, lj]] /.
        Thread[{ln, lj} → ({nn, jj} - LeadingExponent[#])] & /@
        Select[localAnnc, Function[op, With[{lex = LeadingExponent[op]},
          FreeQ[(LeadingExponent[#] - lex + {nn, jj}) & /@ Support[op], {n1_, j1_} /;
            (n1 < 1 || j1 < 0)]]]]] /. c1[nn, jj] → x /. c1 → c], x][[1, 1, 2]]]
  ];

table = Table[c[n, j], {n, 9}, {j, 0, 9}];

Map[If[# === 0, 0, "*"] &, table, {2}] // TableForm

0 * 0 0 0 0 0 0 0 0 0
0 * * 0 0 0 0 0 0 0
0 * * * 0 0 0 0 0
0 * * * * 0 0 0 0
0 * * * * * 0 0 0
0 * * * * * * 0 0
0 * * * * * * * 0
0 * * * * * * * * 0
0 * * * * * * * * *

```

Proof of (Normalization)

In order to get a recurrence that is valid on the diagonal $j=n$, we find an operator whose support consists of power products with $QS[qj, q^j]$ and $QS[qn, q^n]$ having the same power.

```

(* requires 3GB of memory *)
Timing[recDiag = First[FindRelation[annc, OrePolynomialSubstitute → {qj → qn},
  Support → Table[QS[qj, q^j]^i * QS[qn, q^n]^i, {i, 0, 7}]]];]
{32728.1, Null}

recDiag = NormalizeCoefficients[recDiag];

```

Omitting the shifts in j now gives a univariate recurrence of order 7 that is valid on the diagonal $j=n$.

```

recDiag =
  OrePolynomialSubstitute[recDiag, {QS[qj, q^j] → 1}, Algebra → OreAlgebra[QS[qn, q^n]]];
Exponent[OrePolynomialListCoefficients[recDiag], qn]
{84, 84, 88, 88, 92, 92, 92, 92}

```

```

Exponent[OrePolynomialListCoefficients[recDiag], q]
{301, 308, 325, 328, 339, 338, 339, 338}

ByteCount[recDiag]
13 451 904

```

We find that the diagonal recurrence is a multiple of the recurrence of the constant sequence 1.

```

OreReduce[recDiag, Annihilator[1, QS[qn, q^n]]]
0

```

We check whether $c(n,n) = 1$ for $1 \leq n \leq 7$ using our definition of $c(n,j)$:

```

Table[c[n, n], {n, 7}]
{1, 1, 1, 1, 1, 1, 1}

Select[First /@ FactorList[LeadingCoefficient[First[recDiag]]], Exponent[#, qn] === 1 &]
{1 + q^3 qn, 1 + q^4 qn, 1 + q^5 qn, 1 + q^6 qn, 1 + q^7 qn}

```

Together with the fact that the leading coefficient does not have any integer zeros, we have proven that $c(n,n)=1$ for all $n \geq 1$.

Show that we have natural boundaries

We show that our definition of $c(n, j)$ implies $c(n, 0) = 0$ and $c(n, j) = 0$ for $j > n$. These facts will be necessary for the subsequent reasoning, in particular for treating the sums.

The initial values that are required for the following arguments have already been computed (and shown to be 0) when the procedure $c[n,j]$ was defined.

Consider the case $j > n$. Having a look at the support of the first element in $\text{ann}c$

```

Support[annc[[1]]]
{S_{qj,q}^4, S_{qn,q}^2 S_{qj,q}, S_{qj,q}^3, S_{qn,q}^2, S_{qj,q}^2, S_{qj,q}, 1}

```

reveals that we need to prove $c(n, j) = 0$ for the 6 diagonals $j = n+1, \dots, j = n+6$. It then follows that $c(n, j) = 0$ for all $j > n$ by applying this recurrence.

The diagonals are treated in the same way as in the proof of (Normalization).

```

(* a little procedure to plot the monomials in the support of an operator. *)
PlotSupport[p_OrePolynomial] :=
Module[{points, m1, m2},
  points = Reverse[LeadingExponent[#]] & /@ Support[p];
  {m1, m2} = (Max /@ Transpose[points]) + 1;
  Return[ListPlot[points,
    ImageSize -> Round[m1 * 50],
    AspectRatio -> m2 / m1,
    AxesLabel -> ToOrePolynomial[Reverse[OreAlgebraGenerators[OreAlgebra[p]]]],
    PlotMarkers -> Style["x", Bold, 15],
    PlotRange -> {{-1 / 2, m1}, {-1 / 2, m2}},
    Ticks -> {Range[m1] - 1, Range[m2] - 1}]];
];

```

The special case $c(n, 0) = 0$

We first verify that applying the appropriate recurrence from our annihilating ideal implies $c(n, 0) = 0$.

```

LeadingPowerProduct /@ annc
{Sqj,q4, Sqn,q, Sqj,q3, Sqn,q2, Sqj,q2, Sqn,q3, Sqj,q, Sqn,q4}
rec = OrePolynomialSubstitute[annc[[5]], {qj → 1}];
Support[rec]
{Sqn,q4, Sqn,q2, 1}
MyQRoots[LeadingCoefficient[rec], qn]
{0}

```

■ The diagonal $j = n+1$

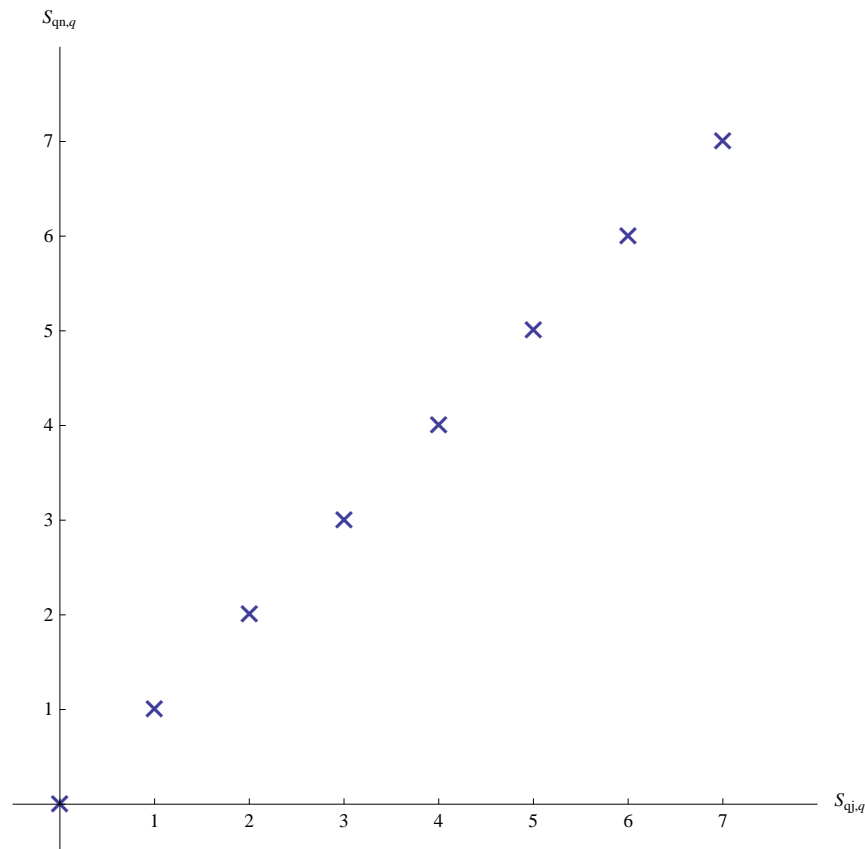
```

Timing[rec = NormalizeCoefficients[First[FindRelation[annc,
OrePolynomialSubstitute → {qj → qn * q},
Support → Table[QS[qj, qj]k * QS[qn, qn]k, {k, 0, 7}],
Modulus → 0]]];]
{24 336.9, Null}

```



```
PlotSupport[rec]
```



```
MyQRroots[LeadingCoefficient[rec], qn]
```

```
{}
```

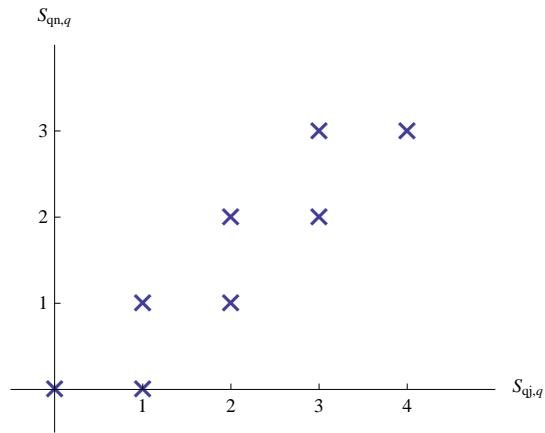
■ The diagonal $j = n+2$

For the diagonal $j = n+2$, we can already make use of the zeros on $j = n+1$. This means that we do not necessarily need an operator whose support lies on one diagonal, but on two diagonals. Therefore the total degree becomes smaller and the computation much easier.

```
Timing[rec = NormalizeCoefficients[First[FindRelation[annc,
  OrePolynomialSubstitute -> {qj -> qn * q}, Support ->
  Flatten[Table[{QS[qj, q^j]^jj * QS[qn, q^n]^nn}, {jj, nn, nn + 1}, {nn, 0, 3}],
  Modulus -> 0]]];]
```

```
{432.851, Null}
```

PlotSupport[rec]



MyQRroots[LeadingCoefficient[rec], qn]

{}

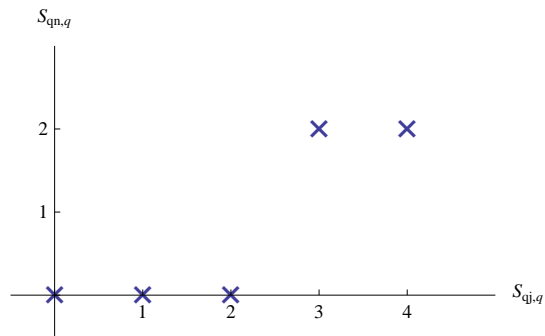
■ The diagonal $j = n+3$

For $j = n+3$, the support of the operator may lie on 3 diagonals, analogously as before.

```
Timing[rec = NormalizeCoefficients[First[FindRelation[anncc,
  OrePolynomialSubstitute -> {qj -> qn * q},
  Support -> (QS[qj, q^j]^#[[1]] * QS[qn, q^n]^#[[2]] & /@
    {{0, 0}, {1, 0}, {2, 0}, {3, 2}, {4, 2}}),
  Modulus -> 0]]];]
```

{126.048, Null}

PlotSupport[rec]



MyQRroots[LeadingCoefficient[rec], qn]

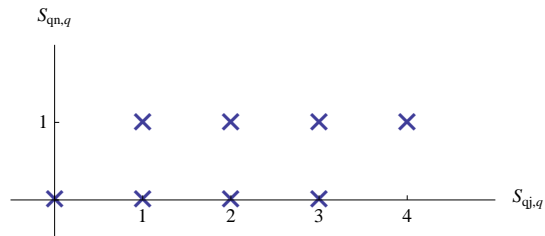
{}

■ The diagonal $j = n+4$

```
Timing[rec = NormalizeCoefficients[First[FindRelation[annc,
  OrePolynomialSubstitute -> {qj -> qn * q}, Support ->
  Flatten[Table[QS[qj, q^j]^jj * QS[qn, q^n]^nn], {jj, nn, nn + 3}, {nn, 0, 1}],
  Modulus -> 0]]];]
```

```
{118.591, Null}
```

```
PlotSupport[rec]
```



```
MyQRroots[LeadingCoefficient[rec], qn]
```

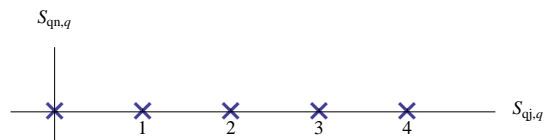
```
{}
```

■ The diagonal $j = n+5$

```
Timing[rec = NormalizeCoefficients[First[FindRelation[annc,
  OrePolynomialSubstitute -> {qj -> qn * q},
  Support -> Table[QS[qj, q^j]^k, {k, 0, 4}],
  Modulus -> 0]]];]
```

```
{23.1534, Null}
```

```
PlotSupport[rec]
```



```
MyQRroots[LeadingCoefficient[rec], qn]
```

```
{}
```

■ The diagonal $j = n+6$

```
Timing[rec = NormalizeCoefficients[Last[FindRelation[annc,
  OrePolynomialSubstitute -> {qj -> qn * q},
  Support -> Table[QS[qj, q^j]^k, {k, 0, 5}],
  Modulus -> 0]]];]
```

```
{56.9956, Null}
```



```
MyQRroots[lcfs[[6]] /. qi -> q /. qn -> q^2, qj]
```

```
Out[24]= {1, 2, 3, 4}
```

```
In[54]:= MyQRroots[lcfs[[6]] /. qi -> q /. qn -> q^3, qj]
```

```
Out[54]= {0, 1, 2, 3, 4}
```

```
In[55]:= MyQRroots[lcfs[[6]] /. qi -> q /. qn -> q^4, qj]
```

```
Out[55]= {0, 1, 2, 3, 4, 5}
```

On the lines $j = 1, 2, 3, 4$ we can apply the second recurrence (except finitely many points near the origin):

```
In[56]:= MyQRroots[lcfs[[2]] /. qi -> q /. qj -> q, qn]
```

```
Out[56]= {0, 1, 2, 3, 4, 5}
```

```
In[57]:= MyQRroots[lcfs[[2]] /. qi -> q /. qj -> q^2, qn]
```

```
Out[57]= {1, 2, 3, 4, 5}
```

```
In[58]:= MyQRroots[lcfs[[2]] /. qi -> q /. qj -> q^3, qn]
```

```
Out[58]= {1, 2, 3, 4, 5}
```

```
In[59]:= MyQRroots[lcfs[[2]] /. qi -> q /. qj -> q^4, qn]
```

```
Out[59]= {1, 2, 3, 4, 5, 6}
```

In the area $i = 1$ and $j > 4$ and $n > 4$, we can safely apply recurrence no. 4 (no singularities here):

```
In[64]:= expts = DeleteCases[Flatten[MapIndexed[(Exponent[#1, q] + {n, j} . (#2 - 1)) &,
      CoefficientList[lcfs[[4]] /. qi -> q, {qn, qj}], {2}]], -Infinity];
```

```
In[65]:= i1 = 1;
While[i1 <= Length[expts],
  i2 = 1;
  While[i2 <= Length[expts] && Simplify[expts[[i1]] < expts[[i2]], n >= 5 && j >= 5] != True, i2++];
  If[i2 > Length[expts], i1++, expts = Drop[expts, {i1}]];
]
```

```
In[67]:= expts
```

```
Out[67]= {18 + 22 j + 18 n, 20 + 21 j + 19 n, 11 + 20 j + 24 n,
  9 + 19 j + 25 n, 13 + 18 j + 26 n, 11 + 17 j + 27 n, 8 + 16 j + 28 n}
```

```
In[68]:= Table[
  {k, " ", Simplify[Reduce[Simplify[n >= 5 && j >= 5 && And@@(expts[[k]] > # & /@ Drop[expts, {k}])]],
  Element[{n, j}, Integers]], {k, Length[expts]}] // TableForm
```

```
Out[68]//TableForm=
```

```
1   n >= 5 && 3 + j >= 3 n
2   False
3   n >= 5 && j >= 2 + n && 4 + j <= 3 n
4   False
5   j <= n && ((j >= 5 && n < 7 && n >= 5) || (n >= 7 && n <= 1 + j))
6   False
7   n > 8 && j >= 5 && 3 + j < n
```

The cases $j=n+1$, $j+2=n$, $j+3=n$ are not contained in the above regions and have to be checked separately:

```
In[69]:= MyQRroots[lcfs[[4]] /. qi -> q /. qj -> q * qn, qn]
```

```
Out[69]= {0, 1, 2, 3}
```

```
In[70]:= MyQRroots[lcfs[[4]] /. qi -> q /. qn -> q^2 * qj, qj]
```

```
Out[70]= {0, 1, 2}
```

```
In[71]:= MyQRroots[lcfs[[4]] /. qi -> q /. qn -> q^3 * qj, qj]
```

```
Out[71]= {0, 1, 2}
```

■ Prove correctness of the certificates

Now we load and assemble the first creative telescoping relation.

```
ansatz = << "ansatz1.m";
denom = << "denom1.m";
solution = << "solution1.m";
ctrel = ansatz /. myDen[a1___][a2_] -> (denom[[a1]] /. qj -> a2) /. solution;

ByteCount[ctrel]

1 479 304 632

reduce = OreReduce[ctrel, ann];
```

We find that reduce is zero, proving that the first creative telescoping relation is correct. The reduction takes 12h and needs 14GB of memory.

A strong indication for the correctness is obtained in much less time by performing the reduction in a modular way. We give some examples in the following

```
Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> 37, qi -> 98, qj -> -113, qn -> -5}]]
{885.555, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> -58, qi -> 20, qj -> 31}]]
{1115.25, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {qi -> 1089, qj -> 81, qn -> 6}]]
{1109.39, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> 67, qn -> 133}]]
{1215.54, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> 13}]]
{1439.91, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {qn -> 27}]]
{1835.44, 0}
```

We now assemble and reduce the second creative telescoping relation.

```

ansatz = << "ansatz2.m";
denom = << "denom2.m";
solution = << "solution2.m";
ctrel = ansatz /. myDen[a1___][a2_] => (denom[[a1]] /. qj -> a2) /. solution;

ByteCount[ctrel]

7 227 816 800

reduce = OreReduce[ctrel, ann];

```

The reduction takes 65h, needs 37GB of memory, and finally yields reduce = 0.

Again, some modular computations that can be carried out more easily and that are nearly equally convincing.

```

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> 3, qi -> 13, qj -> -18, qn -> 47}]]

{2372.38, 0}

Timing[OreReduce[ctrel, ann, OrePolynomialSubstitute -> {q -> -17, qi -> 229, qn -> 42}]]

{2644.46, 0}

```

■ Problem statement

After simple rewriting, (Soichi) looks as follows: For $2 \leq i < n$

```

TraditionalForm[
HoldForm[Sum[q^(i+j-1) * (QBinomial[i+j-2, i-1, q] + q * QBinomial[i+j-1, i, q]) * c[n, j],
{j, 1, n}] = c[n, i-1] - (1+q^i) c[n, i]]]

```

$$\sum_{j=1}^n q^{i+j-1} \left(\binom{i+j-2}{i-1}_q + q \binom{i+j-1}{i}_q \right) c(n, j) = c(n, i-1) - (1+q^i) c(n, i)$$

and for $i = 1$

```

TraditionalForm[
HoldForm[Sum[q^(i+j-1) * (QBinomial[i+j-2, i-1, q] + q * QBinomial[i+j-1, i, q]) * c[n, j],
{j, 1, n}] /. i -> 1 = -(1+q) c[n, 1]]]

```

$$\sum_{j=1}^n q^{i+j-1} \left(\binom{i+j-2}{i-1}_q + q \binom{i+j-1}{i}_q \right) c(n, j) /. i \rightarrow 1 = -(1+q) c(n, 1)$$

Obviously, we can view the case $i=1$ as a special case of the first formula since $c(n,0)=0$ holds (this has been proven before). And this is the reason why we have extended our definition of $c(n,j)$ to $j=0$.

Next, observe that the sum runs over natural boundaries: we have that $c'(n,i,j)=0$ for $j \leq 0$ (because of the binomial coefficients) and that $c'(n,i,j)=0$ for $j > n$ (because $c(n,j)=0$ for $j > n$). This information is sufficient to deduce that when we sum over the creative telescoping relation, the delta part vanishes completely. Hence the above sum is annihilated by either of the two principal parts. We extract them now from the creative telescoping relation.

```

ansatz = << "ansatz1.m"; solution = << "solution1.m";
pp1 = OrePolynomialSubstitute[ansatz, {q[___] -> 0}] /. solution;

ByteCount[pp1]

8 173 952

```

```

Support[pp1]

{Sqi,q3, Sqi,q, Sqn,q2, Sqi,q2, Sqn,q2, Sqi,q, 1}

ansatz = << "ansatz2.m"; solution = << "solution2.m";
pp2 = OrePolynomialSubstitute[ansatz, {q[___] → 0}] /. solution;

ByteCount[pp2]

38 683 336

Support[pp2]

{Sqn,q4, Sqi,q, Sqn,q2, Sqi,q2, Sqn,q2, Sqi,q, 1}

```

■ Determine singularities

In this section we want to show that the two recurrences pp1 and pp2 uniquely define a bivariate sequence, provided that some finite set of initial values is given. In particular we show that they do not have relevant singularities.

```

lcfs = LeadingCoefficient /@ {pp1, pp2};

LeadingPowerProduct /@ {pp1, pp2}

{Sqi,q3, Sqn,q4}

lcfs[[1]] = Numerator[Factor[lcfs[[1]] /. qi → qi / q^3]];
lcfs[[2]] = Numerator[Factor[lcfs[[2]] /. qn → qn / q^4]];

```

The leading coefficients consist of some small factors and one big factor each. We have a look at the small factors and treat the big ones separately.

```

Table[ByteCount /@ (List @@ lcfs[[k]]), {k, 2}]

{{16, 96, 152, 56, 96, 136, 248, 152, 208, 1 251 352}, {56, 96, 152, 96, 152, 152,
  96, 136, 40, 96, 96, 96, 136, 96, 96, 152, 208, 208, 208, 248, 208, 208, 1 251 920}}

```

```

Most[lcfs[[1]]]

- (q - qi) (q2 - qi) (-1 + qi) (qi - qn) (-qi + q qn) (1 + q qn + q2 qn2) (-1 + q qn3) (-1 + q5 qn3)

```

→ is zero for $i=0,1,2$, and $i=n$, and $i=n+1$.

```

Most[lcfs[[2]]]

qi2 (-1 + q qi) (-1 + q2 qi) (q - qn) (q2 - qn) (q3 - qn) (qi - qn)
(q qi - qn) (q + qn) (q2 + qn) (q3 + qn) (q4 + qn) (-qi + q qn) (q - qi qn) (-1 + qi qn)
(q - qn2) (q3 - qn2) (q5 - qn2) (q7 - qn2) (q4 + q2 qn + qn2) (q4 - qn3) (q8 - qn3)

```

→ is zero for $n=1,2,3$, and $i=n$, and $i=n-1$, and $i=n+1$, and $(i,n)=(1,0),(0,1),(0,0)$.

```

f1 = Last[lcfs[[1]]]; f2 = Last[lcfs[[2]]];

```

■ f1 has no zeros

We first test numerically that there are no zeros for $1 \leq i < n \leq 20$:


```

test = f1 /. q -> 2;
Position[Table[test /. qi -> 2^i /. qn -> 2^n, {n, 20}, {i, n - 1}], 0, {2}]
{}

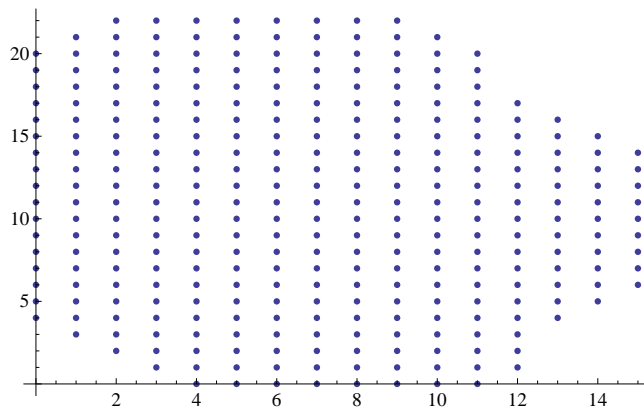
Exponent[f1, {qi, qn}]
{15, 22}

```

In the following we will argue by case distinction that for any choice of integers i and n ($1 \leq i < n$), there is one single term that yields the highest power of q which therefore cannot cancel away. This proves that $f1$ cannot be zero for those choices of i and n .

These are the exponents of q^i and q^n that occur in $f1$:

```
ListPlot[Position[CoefficientList[f1, {qi, qn}], Except[0], {2}, Heads -> False] - 1]
```



Since we are interested in the highest q -powers only, we take for each power product of q^i and q^n only the term with the highest power of q . In the following list we store the corresponding exponents. If we can prove that for any integers i and n there is a single maximal element in this list, we are done.

```

exps = DeleteCases[Flatten[
  MapIndexed[ (#1 + Total[{i, n} * (#2 - 1)]) &,
    Map[Exponent[#, q] &, CoefficientList[f1, {qi, qn}], {2}], {2}]], -Infinity]
{43 + 4 n, 49 + 5 n, 51 + 6 n, 52 + 7 n, 51 + 8 n, 55 + 9 n, 57 + 10 n, 57 + 11 n, 59 + 12 n, 59 + 13 n, 60 + 14 n,
60 + 15 n, 60 + 16 n, 60 + 17 n, 59 + 18 n, 57 + 19 n, 53 + 20 n, 39 + i + 3 n, 46 + i + 4 n, 46 + i + 5 n,
50 + i + 6 n, 51 + i + 7 n, 53 + i + 8 n, 54 + i + 9 n, 55 + i + 10 n, 55 + i + 11 n, 56 + i + 12 n,
55 + i + 13 n, 57 + i + 14 n, 57 + i + 15 n, 57 + i + 16 n, 58 + i + 17 n, 58 + i + 18 n, 57 + i + 19 n,
57 + i + 20 n, 53 + i + 21 n, 35 + 2 i + 2 n, 42 + 2 i + 3 n, 43 + 2 i + 4 n, 44 + 2 i + 5 n, 47 + 2 i + 6 n,
49 + 2 i + 7 n, 49 + 2 i + 8 n, 52 + 2 i + 9 n, 53 + 2 i + 10 n, 52 + 2 i + 11 n, 54 + 2 i + 12 n, 54 + 2 i + 13 n,
55 + 2 i + 14 n, 55 + 2 i + 15 n, 56 + 2 i + 16 n, 56 + 2 i + 17 n, 56 + 2 i + 18 n, 56 + 2 i + 19 n,
55 + 2 i + 20 n, 55 + 2 i + 21 n, 50 + 2 i + 22 n, 32 + 3 i + n, 38 + 3 i + 2 n, 39 + 3 i + 3 n, 42 + 3 i + 4 n,
43 + 3 i + 5 n, 45 + 3 i + 6 n, 47 + 3 i + 7 n, 47 + 3 i + 8 n, 49 + 3 i + 9 n, 51 + 3 i + 10 n, 51 + 3 i + 11 n,
52 + 3 i + 12 n, 53 + 3 i + 13 n, 53 + 3 i + 14 n, 53 + 3 i + 15 n, 53 + 3 i + 16 n, 54 + 3 i + 17 n,
54 + 3 i + 18 n, 54 + 3 i + 19 n, 54 + 3 i + 20 n, 52 + 3 i + 21 n, 52 + 3 i + 22 n, 27 + 4 i, 34 + 4 i + n,
35 + 4 i + 2 n, 38 + 4 i + 3 n, 39 + 4 i + 4 n, 41 + 4 i + 5 n, 43 + 4 i + 6 n, 44 + 4 i + 7 n, 45 + 4 i + 8 n,
47 + 4 i + 9 n, 48 + 4 i + 10 n, 49 + 4 i + 11 n, 48 + 4 i + 12 n, 50 + 4 i + 13 n, 51 + 4 i + 14 n,
50 + 4 i + 15 n, 50 + 4 i + 16 n, 51 + 4 i + 17 n, 51 + 4 i + 18 n, 50 + 4 i + 19 n, 50 + 4 i + 20 n,
51 + 4 i + 21 n, 49 + 4 i + 22 n, 29 + 5 i, 31 + 5 i + n, 33 + 5 i + 2 n, 33 + 5 i + 3 n, 36 + 5 i + 4 n,
38 + 5 i + 5 n, 40 + 5 i + 6 n, 39 + 5 i + 7 n, 42 + 5 i + 8 n, 44 + 5 i + 9 n, 45 + 5 i + 10 n, 46 + 5 i + 11 n,
46 + 5 i + 12 n, 46 + 5 i + 13 n, 48 + 5 i + 14 n, 48 + 5 i + 15 n, 47 + 5 i + 16 n, 48 + 5 i + 17 n, 48 + 5 i + 18 n,
48 + 5 i + 19 n, 48 + 5 i + 20 n, 48 + 5 i + 21 n, 45 + 5 i + 22 n, 26 + 6 i, 28 + 6 i + n,
29 + 6 i + 2 n, 31 + 6 i + 3 n, 33 + 6 i + 4 n, 35 + 6 i + 5 n, 36 + 6 i + 6 n, 38 + 6 i + 7 n, 39 + 6 i + 8 n,
41 + 6 i + 9 n, 42 + 6 i + 10 n, 43 + 6 i + 11 n, 43 + 6 i + 12 n, 43 + 6 i + 13 n, 44 + 6 i + 14 n,
45 + 6 i + 15 n, 45 + 6 i + 16 n, 45 + 6 i + 17 n, 45 + 6 i + 18 n, 45 + 6 i + 19 n, 45 + 6 i + 20 n,
44 + 6 i + 21 n, 42 + 6 i + 22 n, 22 + 7 i, 24 + 7 i + n, 26 + 7 i + 2 n, 28 + 7 i + 3 n, 30 + 7 i + 4 n,
32 + 7 i + 5 n, 34 + 7 i + 6 n, 35 + 7 i + 7 n, 37 + 7 i + 8 n, 38 + 7 i + 9 n, 39 + 7 i + 10 n, 40 + 7 i + 11 n,
41 + 7 i + 12 n, 41 + 7 i + 13 n, 42 + 7 i + 14 n, 42 + 7 i + 15 n, 42 + 7 i + 16 n, 41 + 7 i + 17 n,
42 + 7 i + 18 n, 41 + 7 i + 19 n, 41 + 7 i + 20 n, 41 + 7 i + 21 n, 39 + 7 i + 22 n, 19 + 8 i, 21 + 8 i + n,
23 + 8 i + 2 n, 25 + 8 i + 3 n, 27 + 8 i + 4 n, 29 + 8 i + 5 n, 30 + 8 i + 6 n, 31 + 8 i + 7 n, 33 + 8 i + 8 n,
35 + 8 i + 9 n, 36 + 8 i + 10 n, 37 + 8 i + 11 n, 37 + 8 i + 12 n, 37 + 8 i + 13 n, 38 + 8 i + 14 n,
39 + 8 i + 15 n, 39 + 8 i + 16 n, 39 + 8 i + 17 n, 39 + 8 i + 18 n, 38 + 8 i + 19 n, 37 + 8 i + 20 n,
38 + 8 i + 21 n, 35 + 8 i + 22 n, 16 + 9 i, 18 + 9 i + n, 19 + 9 i + 2 n, 21 + 9 i + 3 n, 23 + 9 i + 4 n,
24 + 9 i + 5 n, 27 + 9 i + 6 n, 28 + 9 i + 7 n, 30 + 9 i + 8 n, 31 + 9 i + 9 n, 32 + 9 i + 10 n, 33 + 9 i + 11 n,
34 + 9 i + 12 n, 34 + 9 i + 13 n, 35 + 9 i + 14 n, 33 + 9 i + 15 n, 35 + 9 i + 16 n, 35 + 9 i + 17 n,
35 + 9 i + 18 n, 32 + 9 i + 19 n, 34 + 9 i + 20 n, 34 + 9 i + 21 n, 31 + 9 i + 22 n, 12 + 10 i, 14 + 10 i + n,
16 + 10 i + 2 n, 18 + 10 i + 3 n, 20 + 10 i + 4 n, 21 + 10 i + 5 n, 22 + 10 i + 6 n, 24 + 10 i + 7 n,
26 + 10 i + 8 n, 27 + 10 i + 9 n, 27 + 10 i + 10 n, 29 + 10 i + 11 n, 30 + 10 i + 12 n, 30 + 10 i + 13 n,
30 + 10 i + 14 n, 31 + 10 i + 15 n, 32 + 10 i + 16 n, 32 + 10 i + 17 n, 30 + 10 i + 18 n, 30 + 10 i + 19 n,
29 + 10 i + 20 n, 30 + 10 i + 21 n, 8 + 11 i, 10 + 11 i + n, 12 + 11 i + 2 n, 14 + 11 i + 3 n, 16 + 11 i + 4 n,
18 + 11 i + 5 n, 19 + 11 i + 6 n, 21 + 11 i + 7 n, 22 + 11 i + 8 n, 23 + 11 i + 9 n, 24 + 11 i + 10 n,
25 + 11 i + 11 n, 26 + 11 i + 12 n, 27 + 11 i + 13 n, 27 + 11 i + 14 n, 28 + 11 i + 15 n, 28 + 11 i + 16 n,
28 + 11 i + 17 n, 28 + 11 i + 18 n, 24 + 11 i + 19 n, 25 + 11 i + 20 n, 5 + 12 i + n, 7 + 12 i + 2 n,
10 + 12 i + 3 n, 12 + 12 i + 4 n, 13 + 12 i + 5 n, 15 + 12 i + 6 n, 15 + 12 i + 7 n, 18 + 12 i + 8 n,
19 + 12 i + 9 n, 20 + 12 i + 10 n, 21 + 12 i + 11 n, 22 + 12 i + 12 n, 21 + 12 i + 13 n, 23 + 12 i + 14 n,
23 + 12 i + 15 n, 24 + 12 i + 16 n, 24 + 12 i + 17 n, 7 + 13 i + 4 n, 9 + 13 i + 5 n, 10 + 13 i + 6 n,
12 + 13 i + 7 n, 13 + 13 i + 8 n, 15 + 13 i + 9 n, 16 + 13 i + 10 n, 17 + 13 i + 11 n, 17 + 13 i + 12 n,
18 + 13 i + 13 n, 18 + 13 i + 14 n, 19 + 13 i + 15 n, 19 + 13 i + 16 n, 4 + 14 i + 5 n, 6 + 14 i + 6 n,
7 + 14 i + 7 n, 9 + 14 i + 8 n, 10 + 14 i + 9 n, 11 + 14 i + 10 n, 12 + 14 i + 11 n, 13 + 14 i + 12 n,
13 + 14 i + 13 n, 14 + 14 i + 14 n, 14 + 14 i + 15 n, 15 i + 6 n, 2 + 15 i + 7 n, 3 + 15 i + 8 n,
5 + 15 i + 9 n, 6 + 15 i + 10 n, 7 + 15 i + 11 n, 7 + 15 i + 12 n, 8 + 15 i + 13 n, 8 + 15 i + 14 n}

```

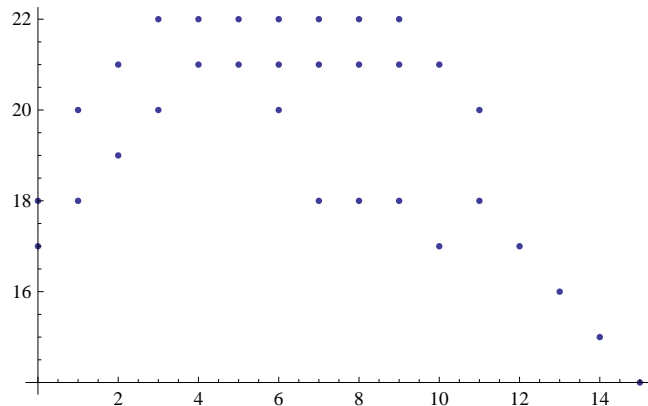
We sort out all elements that are obviously smaller than some other for all $i > 0$ and $n > 0$.

```

extremal = exps;
Function[expl,
  c1 = Flatten[PadRight[CoefficientList[expl, {i, n}], {2, 2}]]; extremal = DeleteCases[
    extremal, exp2_ /; With[{c2 = Flatten[PadRight[CoefficientList[exp2, {i, n}], {2, 2}]}],
      (And@@Thread[c2 ≤ c1]) && Total[c2] < Total[c1]]];] /@ exps;
extremal
{60 + 17 n, 59 + 18 n, 58 + i + 18 n, 57 + i + 20 n, 56 + 2 i + 19 n, 55 + 2 i + 21 n, 54 + 3 i + 20 n,
52 + 3 i + 22 n, 51 + 4 i + 21 n, 49 + 4 i + 22 n, 48 + 5 i + 21 n, 45 + 5 i + 22 n, 45 + 6 i + 20 n, 44 + 6 i + 21 n,
42 + 6 i + 22 n, 42 + 7 i + 18 n, 41 + 7 i + 21 n, 39 + 7 i + 22 n, 39 + 8 i + 18 n, 38 + 8 i + 21 n,
35 + 8 i + 22 n, 35 + 9 i + 18 n, 34 + 9 i + 21 n, 31 + 9 i + 22 n, 32 + 10 i + 17 n, 30 + 10 i + 21 n,
28 + 11 i + 18 n, 25 + 11 i + 20 n, 24 + 12 i + 17 n, 19 + 13 i + 16 n, 14 + 14 i + 15 n, 8 + 15 i + 14 n}

```

```
ListPlot[Coefficient[#, {i, n}] & /@ extremal]
```



We compare all terms with the same n-coefficient and figure out the minimal i_0 such that the right-most term (i.e., the one with the largest i-coefficient) always takes the biggest value among these terms for all $i > i_0$.

```
Do[
  With[{row = Select[extremal, Coefficient[#, n] === nn &] /. n -> 0},
    Print["nn = ", nn, ", i > ", Max[Solve[# == Last[row], i][[1, 1, 2]] & /@ Most[row]]];]
  ,
  {nn,
   14,
   22}]
```

```
nn = 14, i > -∞
```

```
nn = 15, i > -∞
```

```
nn = 16, i > -∞
```

```
nn = 17, i > 4
```

```
nn = 18, i >  $\frac{11}{3}$ 
```

```
nn = 19, i > -∞
```

```
nn = 20, i > 4
```

```
nn = 21, i > 4
```

```
nn = 22, i > 4
```

Provided that $i > 4$ we can restrict our investigation to the following set of exponents:

```
extremal = Table[Last[Select[extremal, Coefficient[#, n] === nn &]], {nn, 14, 22}]
```

```
{8 + 15 i + 14 n, 14 + 14 i + 15 n, 19 + 13 i + 16 n, 24 + 12 i + 17 n,
 28 + 11 i + 18 n, 56 + 2 i + 19 n, 25 + 11 i + 20 n, 30 + 10 i + 21 n, 31 + 9 i + 22 n}
```

TableForm[Table[Position[extremal, Max[extremal]], {i, 0, 15}, {n, 0, 15}]]

6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
6	6	6	6	6	6	$\frac{6}{9}$	9	9	9	9	9	9	9	9
6	6	6	6	9	9	9	9	9	9	9	9	9	9	9
6	$\frac{6}{8}$	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9	9	9
4	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9	9
5	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9	9
2	4	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9
3	5	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9
4	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9	9
1	2	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9
2	3	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9
	4	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9	9
		2	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9
		3	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9
1	1	4	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9
		7	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9
		8	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9	9
1	1	1	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9	9
1	1	1	$\frac{1}{2}$	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9	9	9	9	9
1	1	1	1	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9	9	9	9
1	1	1	1	1	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9	9	9
1	1	1	1	1	1	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9	9
1	1	1	1	1	1	7	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9	9
1	1	1	1	1	1	1	7	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$	9
1	1	1	1	1	1	1	1	7	7	$\frac{7}{8}$	8	8	8	$\frac{8}{9}$

It is now easy to show that for n big enough the term $q^{31}(q^i)^9(q^n)^{22}$ will be the highest power of q in f1, for $4 < i < n$.

FullSimplify[And@@((Last[extremal] > #) & /@Most[extremal]), 4 < i < n && n ≥ 9]

True

For $n < 9$ there is nothing to do, since these (finitely many; note that $n < 9$ implies $i < 8$) cases have been checked (numerically) at the beginning of this section.

We check the cases $i=1,2,3,4$ separately. After plugging in the corresponding q-power for q^i , we are left with a bivariate polynomial in q and q^n .

MyQRroots[f1 /. qi → q, qn]

{0}

MyQRroots[f1 /. qi → q^2, qn]

{0, 1}

MyQRroots[f1 /. qi → q^3, qn]

{0, 1}

```
MyQRoots[f1 /. qi -> q^4, qn]
{0}
```

- **f2 has zeros at (i,n)=(1,2),(1,3),(1,4),(2,4),(3,4),(3,5)**

We do a modular check and find all candidates for zeros in the area $1 \leq i < n \leq 20$:

```
test = f2 /. q -> 2;
Reverse /@ Position[Table[test /. qi -> 2^i /. qn -> 2^n, {n, 20}, {i, n - 1}], 0, {2}]
{{1, 2}, {1, 3}, {1, 4}, {2, 4}, {3, 4}, {3, 5}}

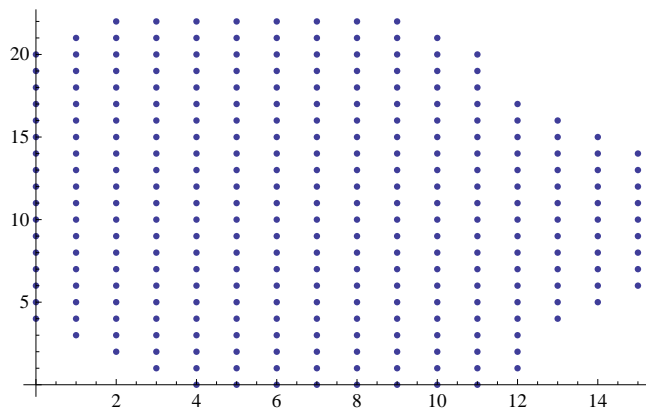
f2 /. Thread[{qi, qn} -> q^#] & /@ %
{0, 0, 0, 0, 0, 0}

Exponent[f2, {qi, qn}]
{15, 22}
```

In the following we will argue by case distinction that for any choice of integers i and n ($1 \leq i < n$), there is one single term that yields the highest power of q which therefore cannot cancel away. This proves that $f2$ cannot be zero for those choices of i and n .

These are the exponents of q^i and q^n that occur in $f2$:

```
ListPlot[Position[CoefficientList[f2, {qi, qn}], Except[0], {2}, Heads -> False] - 1]
```



Since we are interested in the highest q -powers only, we take for each power product of q^i and q^n only the term with the highest power of q . In the following list we store the corresponding exponents. If we can prove that for any integers i and n there is a single maximal element in this list, we are done.

```

exps = DeleteCases[Flatten[
  MapIndexed[(#1 + Total[{i, n} * (#2 - 1)]) &,
    Map[Exponent[#, q] &, CoefficientList[f2, {qi, qn}], {2}], {2}]], -Infinity]
{66 + 4 n, 68 + 5 n, 66 + 6 n, 63 + 7 n, 58 + 8 n, 58 + 9 n, 56 + 10 n, 52 + 11 n, 50 + 12 n, 46 + 13 n, 43 + 14 n,
  39 + 15 n, 35 + 16 n, 31 + 17 n, 26 + 18 n, 20 + 19 n, 12 + 20 n, 69 + i + 3 n, 72 + i + 4 n, 68 + i + 5 n,
  68 + i + 6 n, 65 + i + 7 n, 63 + i + 8 n, 60 + i + 9 n, 57 + i + 10 n, 53 + i + 11 n, 50 + i + 12 n,
  45 + i + 13 n, 43 + i + 14 n, 39 + i + 15 n, 35 + i + 16 n, 32 + i + 17 n, 28 + i + 18 n, 23 + i + 19 n,
  19 + i + 20 n, 11 + i + 21 n, 72 + 2 i + 2 n, 75 + 2 i + 3 n, 72 + 2 i + 4 n, 69 + 2 i + 5 n, 68 + 2 i + 6 n,
  66 + 2 i + 7 n, 62 + 2 i + 8 n, 61 + 2 i + 9 n, 58 + 2 i + 10 n, 53 + 2 i + 11 n, 51 + 2 i + 12 n, 47 + 2 i + 13 n,
  44 + 2 i + 14 n, 40 + 2 i + 15 n, 37 + 2 i + 16 n, 33 + 2 i + 17 n, 29 + 2 i + 18 n, 25 + 2 i + 19 n,
  20 + 2 i + 20 n, 16 + 2 i + 21 n, 7 + 2 i + 22 n, 76 + 3 i + n, 78 + 3 i + 2 n, 75 + 3 i + 3 n, 74 + 3 i + 4 n,
  71 + 3 i + 5 n, 69 + 3 i + 6 n, 67 + 3 i + 7 n, 63 + 3 i + 8 n, 61 + 3 i + 9 n, 59 + 3 i + 10 n, 55 + 3 i + 11 n,
  52 + 3 i + 12 n, 49 + 3 i + 13 n, 45 + 3 i + 14 n, 41 + 3 i + 15 n, 37 + 3 i + 16 n, 34 + 3 i + 17 n,
  30 + 3 i + 18 n, 26 + 3 i + 19 n, 22 + 3 i + 20 n, 16 + 3 i + 21 n, 12 + 3 i + 22 n, 78 + 4 i, 81 + 4 i + n,
  78 + 4 i + 2 n, 77 + 4 i + 3 n, 74 + 4 i + 4 n, 72 + 4 i + 5 n, 70 + 4 i + 6 n, 67 + 4 i + 7 n, 64 + 4 i + 8 n,
  62 + 4 i + 9 n, 59 + 4 i + 10 n, 56 + 4 i + 11 n, 51 + 4 i + 12 n, 49 + 4 i + 13 n, 46 + 4 i + 14 n,
  41 + 4 i + 15 n, 37 + 4 i + 16 n, 34 + 4 i + 17 n, 30 + 4 i + 18 n, 25 + 4 i + 19 n, 21 + 4 i + 20 n,
  18 + 4 i + 21 n, 12 + 4 i + 22 n, 83 + 5 i, 81 + 5 i + n, 79 + 5 i + 2 n, 75 + 5 i + 3 n, 74 + 5 i + 4 n,
  72 + 5 i + 5 n, 70 + 5 i + 6 n, 65 + 5 i + 7 n, 64 + 5 i + 8 n, 62 + 5 i + 9 n, 59 + 5 i + 10 n, 56 + 5 i + 11 n,
  52 + 5 i + 12 n, 48 + 5 i + 13 n, 46 + 5 i + 14 n, 42 + 5 i + 15 n, 37 + 5 i + 16 n, 34 + 5 i + 17 n,
  30 + 5 i + 18 n, 26 + 5 i + 19 n, 22 + 5 i + 20 n, 18 + 5 i + 21 n, 11 + 5 i + 22 n, 83 + 6 i, 81 + 6 i + n,
  78 + 6 i + 2 n, 76 + 6 i + 3 n, 74 + 6 i + 4 n, 72 + 6 i + 5 n, 69 + 6 i + 6 n, 67 + 6 i + 7 n, 64 + 6 i + 8 n,
  62 + 6 i + 9 n, 59 + 6 i + 10 n, 56 + 6 i + 11 n, 52 + 6 i + 12 n, 48 + 6 i + 13 n, 45 + 6 i + 14 n,
  42 + 6 i + 15 n, 38 + 6 i + 16 n, 34 + 6 i + 17 n, 30 + 6 i + 18 n, 26 + 6 i + 19 n, 22 + 6 i + 20 n,
  17 + 6 i + 21 n, 11 + 6 i + 22 n, 82 + 7 i, 80 + 7 i + n, 78 + 7 i + 2 n, 76 + 7 i + 3 n, 74 + 7 i + 4 n,
  72 + 7 i + 5 n, 70 + 7 i + 6 n, 67 + 7 i + 7 n, 65 + 7 i + 8 n, 62 + 7 i + 9 n, 59 + 7 i + 10 n, 56 + 7 i + 11 n,
  53 + 7 i + 12 n, 49 + 7 i + 13 n, 46 + 7 i + 14 n, 42 + 7 i + 15 n, 38 + 7 i + 16 n, 33 + 7 i + 17 n,
  30 + 7 i + 18 n, 25 + 7 i + 19 n, 21 + 7 i + 20 n, 17 + 7 i + 21 n, 11 + 7 i + 22 n, 82 + 8 i, 80 + 8 i + n,
  78 + 8 i + 2 n, 76 + 8 i + 3 n, 74 + 8 i + 4 n, 72 + 8 i + 5 n, 69 + 8 i + 6 n, 66 + 8 i + 7 n, 64 + 8 i + 8 n,
  62 + 8 i + 9 n, 59 + 8 i + 10 n, 56 + 8 i + 11 n, 52 + 8 i + 12 n, 48 + 8 i + 13 n, 45 + 8 i + 14 n,
  42 + 8 i + 15 n, 38 + 8 i + 16 n, 34 + 8 i + 17 n, 30 + 8 i + 18 n, 25 + 8 i + 19 n, 20 + 8 i + 20 n,
  17 + 8 i + 21 n, 10 + 8 i + 22 n, 82 + 9 i, 80 + 9 i + n, 77 + 9 i + 2 n, 75 + 9 i + 3 n, 73 + 9 i + 4 n,
  70 + 9 i + 5 n, 69 + 9 i + 6 n, 66 + 9 i + 7 n, 64 + 9 i + 8 n, 61 + 9 i + 9 n, 58 + 9 i + 10 n, 55 + 9 i + 11 n,
  52 + 9 i + 12 n, 48 + 9 i + 13 n, 45 + 9 i + 14 n, 39 + 9 i + 15 n, 37 + 9 i + 16 n, 33 + 9 i + 17 n,
  29 + 9 i + 18 n, 22 + 9 i + 19 n, 20 + 9 i + 20 n, 16 + 9 i + 21 n, 9 + 9 i + 22 n, 81 + 10 i, 79 + 10 i + n,
  77 + 10 i + 2 n, 75 + 10 i + 3 n, 73 + 10 i + 4 n, 70 + 10 i + 5 n, 67 + 10 i + 6 n, 65 + 10 i + 7 n,
  63 + 10 i + 8 n, 60 + 10 i + 9 n, 56 + 10 i + 10 n, 54 + 10 i + 11 n, 51 + 10 i + 12 n, 47 + 10 i + 13 n,
  43 + 10 i + 14 n, 40 + 10 i + 15 n, 37 + 10 i + 16 n, 33 + 10 i + 17 n, 27 + 10 i + 18 n, 23 + 10 i + 19 n,
  18 + 10 i + 20 n, 15 + 10 i + 21 n, 80 + 11 i, 78 + 11 i + n, 76 + 11 i + 2 n, 74 + 11 i + 3 n, 72 + 11 i + 4 n,
  70 + 11 i + 5 n, 67 + 11 i + 6 n, 65 + 11 i + 7 n, 62 + 11 i + 8 n, 59 + 11 i + 9 n, 56 + 11 i + 10 n,
  53 + 11 i + 11 n, 50 + 11 i + 12 n, 47 + 11 i + 13 n, 43 + 11 i + 14 n, 40 + 11 i + 15 n, 36 + 11 i + 16 n,
  32 + 11 i + 17 n, 28 + 11 i + 18 n, 20 + 11 i + 19 n, 17 + 11 i + 20 n, 76 + 12 i + n, 74 + 12 i + 2 n,
  73 + 12 i + 3 n, 71 + 12 i + 4 n, 68 + 12 i + 5 n, 66 + 12 i + 6 n, 62 + 12 i + 7 n, 61 + 12 i + 8 n,
  58 + 12 i + 9 n, 55 + 12 i + 10 n, 52 + 12 i + 11 n, 49 + 12 i + 12 n, 44 + 12 i + 13 n, 42 + 12 i + 14 n,
  38 + 12 i + 15 n, 35 + 12 i + 16 n, 31 + 12 i + 17 n, 69 + 13 i + 4 n, 67 + 13 i + 5 n, 64 + 13 i + 6 n,
  62 + 13 i + 7 n, 59 + 13 i + 8 n, 57 + 13 i + 9 n, 54 + 13 i + 10 n, 51 + 13 i + 11 n, 47 + 13 i + 12 n,
  44 + 13 i + 13 n, 40 + 13 i + 14 n, 37 + 13 i + 15 n, 33 + 13 i + 16 n, 65 + 14 i + 5 n, 63 + 14 i + 6 n,
  60 + 14 i + 7 n, 58 + 14 i + 8 n, 55 + 14 i + 9 n, 52 + 14 i + 10 n, 49 + 14 i + 11 n, 46 + 14 i + 12 n,
  42 + 14 i + 13 n, 39 + 14 i + 14 n, 35 + 14 i + 15 n, 60 + 15 i + 6 n, 58 + 15 i + 7 n, 55 + 15 i + 8 n,
  53 + 15 i + 9 n, 50 + 15 i + 10 n, 47 + 15 i + 11 n, 43 + 15 i + 12 n, 40 + 15 i + 13 n, 36 + 15 i + 14 n}

```

We sort out all elements that are obviously smaller than some other for all $i > 0$ and $n > 0$.

```

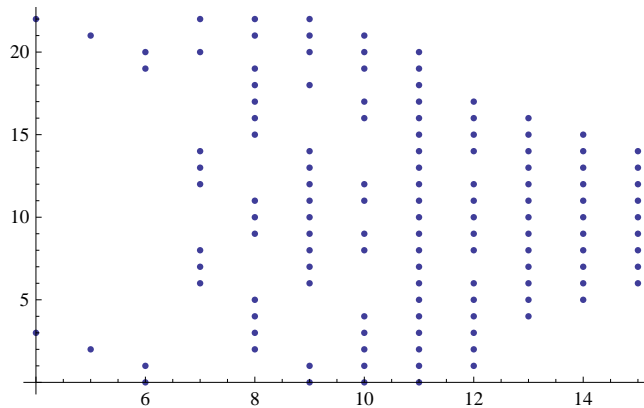
extremal = exps;
Function[expl,
  c1 = Flatten[PadRight[CoefficientList[expl, {i, n}], {2, 2}]]; extremal = DeleteCases[
  extremal, exp2_ /; With[{c2 = Flatten[PadRight[CoefficientList[exp2, {i, n}], {2, 2}]]},
    (And@@Thread[c2 ≤ c1]) && Total[c2] < Total[c1]];] /@ exps;

```

extremal

```
{77 + 4 i + 3 n, 12 + 4 i + 22 n, 79 + 5 i + 2 n, 18 + 5 i + 21 n, 83 + 6 i, 81 + 6 i + n, 26 + 6 i + 19 n,
22 + 6 i + 20 n, 70 + 7 i + 6 n, 67 + 7 i + 7 n, 65 + 7 i + 8 n, 53 + 7 i + 12 n, 49 + 7 i + 13 n,
46 + 7 i + 14 n, 21 + 7 i + 20 n, 11 + 7 i + 22 n, 78 + 8 i + 2 n, 76 + 8 i + 3 n, 74 + 8 i + 4 n,
72 + 8 i + 5 n, 62 + 8 i + 9 n, 59 + 8 i + 10 n, 56 + 8 i + 11 n, 42 + 8 i + 15 n, 38 + 8 i + 16 n,
34 + 8 i + 17 n, 30 + 8 i + 18 n, 25 + 8 i + 19 n, 17 + 8 i + 21 n, 10 + 8 i + 22 n, 82 + 9 i, 80 + 9 i + n,
69 + 9 i + 6 n, 66 + 9 i + 7 n, 64 + 9 i + 8 n, 61 + 9 i + 9 n, 58 + 9 i + 10 n, 55 + 9 i + 11 n,
52 + 9 i + 12 n, 48 + 9 i + 13 n, 45 + 9 i + 14 n, 29 + 9 i + 18 n, 20 + 9 i + 20 n, 16 + 9 i + 21 n,
9 + 9 i + 22 n, 81 + 10 i, 79 + 10 i + n, 77 + 10 i + 2 n, 75 + 10 i + 3 n, 73 + 10 i + 4 n,
63 + 10 i + 8 n, 60 + 10 i + 9 n, 54 + 10 i + 11 n, 51 + 10 i + 12 n, 37 + 10 i + 16 n, 33 + 10 i + 17 n,
23 + 10 i + 19 n, 18 + 10 i + 20 n, 15 + 10 i + 21 n, 80 + 11 i, 78 + 11 i + n, 76 + 11 i + 2 n,
74 + 11 i + 3 n, 72 + 11 i + 4 n, 70 + 11 i + 5 n, 67 + 11 i + 6 n, 65 + 11 i + 7 n, 62 + 11 i + 8 n,
59 + 11 i + 9 n, 56 + 11 i + 10 n, 53 + 11 i + 11 n, 50 + 11 i + 12 n, 47 + 11 i + 13 n, 43 + 11 i + 14 n,
40 + 11 i + 15 n, 36 + 11 i + 16 n, 32 + 11 i + 17 n, 28 + 11 i + 18 n, 20 + 11 i + 19 n, 17 + 11 i + 20 n,
76 + 12 i + n, 74 + 12 i + 2 n, 73 + 12 i + 3 n, 71 + 12 i + 4 n, 68 + 12 i + 5 n, 66 + 12 i + 6 n,
61 + 12 i + 8 n, 58 + 12 i + 9 n, 55 + 12 i + 10 n, 52 + 12 i + 11 n, 49 + 12 i + 12 n, 42 + 12 i + 14 n,
38 + 12 i + 15 n, 35 + 12 i + 16 n, 31 + 12 i + 17 n, 69 + 13 i + 4 n, 67 + 13 i + 5 n, 64 + 13 i + 6 n,
62 + 13 i + 7 n, 59 + 13 i + 8 n, 57 + 13 i + 9 n, 54 + 13 i + 10 n, 51 + 13 i + 11 n, 47 + 13 i + 12 n,
44 + 13 i + 13 n, 40 + 13 i + 14 n, 37 + 13 i + 15 n, 33 + 13 i + 16 n, 65 + 14 i + 5 n, 63 + 14 i + 6 n,
60 + 14 i + 7 n, 58 + 14 i + 8 n, 55 + 14 i + 9 n, 52 + 14 i + 10 n, 49 + 14 i + 11 n, 46 + 14 i + 12 n,
42 + 14 i + 13 n, 39 + 14 i + 14 n, 35 + 14 i + 15 n, 60 + 15 i + 6 n, 58 + 15 i + 7 n, 55 + 15 i + 8 n,
53 + 15 i + 9 n, 50 + 15 i + 10 n, 47 + 15 i + 11 n, 43 + 15 i + 12 n, 40 + 15 i + 13 n, 36 + 15 i + 14 n}
```

ListPlot[Coefficient[#, {i, n}] & /@ extremal]



We compare all terms with the same n -coefficient and figure out the minimal i_0 such that the right-most term (i.e., the one with the largest i -coefficient) always takes the biggest value among these terms for all $i > i_0$.

```
Do[
  With[{row = Select[extremal, Coefficient[#, n] == nn &] /. n -> 0},
    Print["nn = ", nn, ", i > ", Max[Solve[# == Last[row], i][[1, 1, 2]] & /@ Most[row]]];]
  ,
  {nn,
  0,
  22}]
```

```

nn = 0, i > 1
nn = 1, i > 2
nn = 2, i > 2
nn = 3, i > 1
nn = 4, i > 2
nn = 5, i > 2
nn = 6, i > 3
nn = 7, i > 2
nn = 8, i > 3
nn = 9, i > 2
nn = 10, i > 2
nn = 11, i > 2
nn = 12, i > 3
nn = 13, i > 2
nn = 14, i > 3
nn = 15, i > 2
nn = 16, i > 2
nn = 17, i > 1
nn = 18, i >  $\frac{2}{3}$ 
nn = 19, i > 3
nn = 20, i >  $\frac{3}{2}$ 
nn = 21, i > 1
nn = 22, i > 1

```

Provided that $i > 3$ we can restrict our investigation to the following set of exponents:

```

extremal = Table[Last[Select[extremal, Coefficient[#, n] === nn &]], {nn, 0, 22}]
{80 + 11 i, 76 + 12 i + n, 74 + 12 i + 2 n, 73 + 12 i + 3 n, 69 + 13 i + 4 n, 65 + 14 i + 5 n,
60 + 15 i + 6 n, 58 + 15 i + 7 n, 55 + 15 i + 8 n, 53 + 15 i + 9 n, 50 + 15 i + 10 n, 47 + 15 i + 11 n,
43 + 15 i + 12 n, 40 + 15 i + 13 n, 36 + 15 i + 14 n, 35 + 14 i + 15 n, 33 + 13 i + 16 n,
31 + 12 i + 17 n, 28 + 11 i + 18 n, 20 + 11 i + 19 n, 17 + 11 i + 20 n, 15 + 10 i + 21 n, 9 + 9 i + 22 n}

```


TableForm[Table[Position[extremal, Max[extremal]], {i, 0, 15}, {n, 0, 15}]]

1	1	1	4 18 19	19	22	22 23	23	23	23	23	23	23	23	23	23
1	1	1 4	10 11 12	18 19	22	22	22 23	23	23	23	23	23	23	23	23
1	1	4 5 6	10 11 12	16 17 18	18 19 22	22	22	22 23	23	23	23	23	23	23	23
1	1 2	6 7 8	10 11 12	14 15 16	16 17 18	22	22	22	22 23	23	23	23	23	23	23
1 2	6 7	7 8	10 11 12	14 15	15 16	16 17 18 21 22	22	22	22	22 23	23	23	23	23	23
2	7	7 8	10 11 12	14 15	15 16	15 16	21 22	22	22	22	22	22 23	23	23	23
7	7	7 8	10 11 12	14 15	15	15	15 16	21 22	22	22	22	22	22 23	23	23
7	7	7 8	10 11 12	14 15	15	15	15	21	21 22	22	22	22	22	22 23	23
7	7	7 8	10 11 12	14 15	15	15	15	15	21	21 22	22	22	22	22	22 23
7	7	7 8	10 11 12	14 15	15	15	15	15	15	21	21 22	22	22	22	22 23
7	7	7 8	10 11 12	14 15	15	15	15	15	15	15	21	21	21 22	22	22
7	7	7 8	10 11 12	14 15	15	15	15	15	15	15	15	15	21	21	21 22
7	7	7 8	10 11 12	14 15	15	15	15	15	15	15	15	15	15	21	21
7	7	7 8	10 11 12	14 15	15	15	15	15	15	15	15	15	15	15	21

It is now easy to show that for $3 < i \leq n-7$ the term $q^{9*(q^i)^9*(q^n)^{22}}$ will be the highest power of q in f_2 .

FullSimplify[And@@((Last[extremal] > #) & /@Most[extremal]), 3 < i ≤ n - 7]

True

We check the cases $i=1, 2, 3,$ and $i=n-6, n-5, \dots, n-1$ separately. After plugging in the corresponding q -power for q^i , we are left with a polynomial in q and q^n only.

```
MyQRroots[f2 /. qi -> q, qn]
```

```
{2, 3, 4}
```

→ gives zeros at (i,n)=(1,2),(1,3),(1,4)

```
MyQRroots[f2 /. qi -> q^2, qn]
```

```
{2, 4}
```

→ gives zeros at (i,n)=(2,4)

```
MyQRroots[f2 /. qi -> q^3, qn]
```

```
{4, 5}
```

→ gives zeros at (i,n)=(3,4),(3,5)

```
MyQRroots[f2 /. qi -> qn / q^6, qn]
```

```
{4, 5}
```

```
MyQRroots[f2 /. qi -> qn / q^5, qn]
```

```
{2, 3, 4, 5}
```

```
MyQRroots[f2 /. qi -> qn / q^4, qn]
```

```
{2, 3, 4}
```

```
MyQRroots[f2 /. qi -> qn / q^3, qn]
```

```
{2, 3, 4}
```

→ gives a zero at (i,n)=(1,4)

```
MyQRroots[f2 /. qi -> qn / q^2, qn]
```

```
{2, 3, 4, 5}
```

→ gives zeros at (i,n)=(1,3),(2,4),(3,5)

```
MyQRroots[f2 /. qi -> qn / q, qn]
```

```
{1, 2, 4}
```

→ gives zeros at (i,n)=(1,2),(3,4)

Summarizing, the zeros of f_2 are (1,2),(1,3),(1,4),(2,4),(3,4),(3,5), and these are exactly the ones found at the beginning.

■ Conclusion

```
Support [pp1]
```

```
{Sqi,q3, Sqi,q Sqn,q2, Sqi,q2, Sqn,q2, Sqi,q, 1}
```

```
Support [pp2]
```

```
{Sqn,q4, Sqi,q Sqn,q2, Sqi,q2, Sqn,q2, Sqi,q, 1}
```

Together with the constraints $1 \leq i < n$, these supports imply that we have to give initial values for $(i=1 \text{ and } 2 \leq n \leq 7)$, $(i=2 \text{ and } 3 \leq n \leq 8)$, $(i=3 \text{ and } 4 \leq n \leq 9)$.

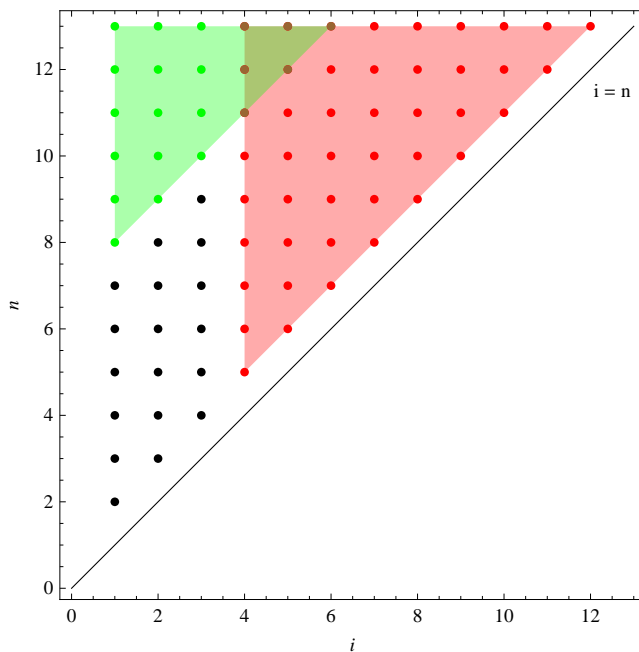
For $i \leq 3$, the first recurrence cannot be used. The leading coefficient of pp2 does not have zeros in this area that are not already part of the initial conditions.

On the lines $i=n-1$, $i=n-2$, ..., $i=n-6$, the second recurrence cannot be used. The leading coefficient of pp1 does not have zeros here.

There are no common zeros of the two leading coefficients that could be relevant for the remaining area.

The situation is depicted in the picture below. Red are the points where the first recurrence can be applied, the area where the second can be used is shown in green. The black points are the ones that have to be given as initial conditions.

```
Show[Graphics[Join[
  {Line[{{0, 0}, {13, 13}], Text["i = n", {25 / 2, 23 / 2}]},
  {Opacity[1 / 3], Red, Polygon[{{4, 5}, {12, 13}, {4, 13}]},
  {Green, Polygon[{{1, 8}, {6, 13}, {1, 13}]}], {Opacity[1], Red},
  Flatten[Table[Disk[{x, y}, 1 / 10], {x, 4, 12}, {y, x + 1, 13}]],
  {Green},
  Flatten[Table[Disk[{x, y}, 1 / 10], {x, 3}, {y, x + 7, 13}]],
  {Brown},
  Flatten[Table[Disk[{x, y}, 1 / 10], {x, 4, 6}, {y, x + 7, 13}]],
  {Black},
  Flatten[Table[Disk[{x, y}, 1 / 10], {x, 3}, {y, x + 1, x + 6}]]
], Frame -> True, FrameLabel -> {i, n}]]
```



■ The right-hand side

We compute an annihilating ideal for the right-hand side and show that pp1 and pp2 are contained in it.

```
(* Timing for DFiniteSubstitute is 1h, for DFiniteOreAction it is 8h. *)
nm = $NormalizeMethod; $NormalizeMethod = "gcd"; (* causes a speed-up in this instance *)
Timing[rhs = DFiniteOreAction[DFiniteSubstitute[annc, {j -> i - 1},
  Algebra -> OreAlgebra[QS[qn, q^n], QS[qi, q^i]]], 1 - (1 + qi) * QS[qi, q^i]]];

(* takes 63s. red1 = 0. *)
Timing[red1 = OreReduce[pp1, rhs];]

(* takes 16h and 4GB of memory. red2 = 0.*)
Timing[red2 = OreReduce[pp2, rhs];]
$NormalizeMethod = nm;
```

We compare initial values for ($i=1$ and $2 \leq n \leq 7$), ($i=2$ and $3 \leq n \leq 8$), ($i=3$ and $4 \leq n \leq 9$), see the subsection on singularities above. Again, we do not treat the case $i=1$ separately, instead we take the degenerate right-hand side into account via $c(n,0)=0$.

```
Table[
  Sum[FunctionExpand[q^(i + j - 1) * (QBinomial[i + j - 2, i - 1, q] + q * QBinomial[i + j - 1, i, q])] *
    c[n, j], {j, n}] == c[n, i - 1] - (1 + q^i) c[n, i],
  {n, i + 1, i + 6}, {i, 3}] // Simplify

{{True, True, True}, {True, True, True}, {True, True, True},
 {True, True, True}, {True, True, True}, {True, True, True}}
```

This completes the proof of (Soichi). Quod erat demonstrandum.

Proof of (Okada)

We rewrite the sum in (Okada) slightly:

```
TraditionalForm[HoldForm[
  Sum[q^(n + j - 1) * (q^(n + j) + q^n - q - 1) / (q^n - 1) * QBinomial[n + j - 2, n - 1, q] * c[n, j],
    {j, 1, n}] + (q^n + 1) - c[n, n - 1] == b[n] / b[n - 1]]]


$$\sum_{j=1}^n \frac{q^{n+j-1} (q^{n+j} + q^n - q - 1) \binom{n+j-2}{n-1}_q c(n, j)}{q^n - 1} + (q^n + 1) - c(n, n-1) = \frac{b(n)}{b(n-1)}$$

```

We start by computing an annihilating ideal for the summand:

```
anj = Annihilator[q^(n + j - 1) * (q^(n + j) + q^n - q - 1) / (q^n - 1) * QBinomial[n + j - 2, n - 1, q],
  {QS[qn, q^n], QS[qj, q^j]}]

{(-1 - q + qj + q qj + qn - qj^2 qn) Sqj, q + (q + q^2 - q qn - qj qn - q qj qn - q^2 qj qn + qj qn^2 + q qj^2 qn^2),
 (-1 - q + qn + q qn + q^2 qn + qj qn - q qn^2 - q qj qn^2) Sqn, q +
 (q + q^2 - q^2 qn - qj qn - q qj qn - q^2 qj qn + q qj qn^2 + q qj^2 qn^2)}
```

```
(* takes about 3 hours. *)
ann = DFiniteTimes[annc, anj];
```

■ Singularity analysis of the annihilating ideal for the summand

We show that there are only finitely many singular points, which are additionally located close to the origin.

```
lcfs = Numerator[Together[
  (LeadingCoefficient[#] /. Thread[{qn, qj} -> {qn, qj} / q^LeadingExponent[#]]) & /@ ann]];
```

```
In[21]:= LeadingPowerProduct /@ ann
```

```
Out[21]= {Sqj,q4, Sqn,q Sqj,q3, Sqn,q2 Sqj,q2, Sqn,q3 Sqj,q, Sqn,q4}
```

On the lines $n = 1, 2, 3, 4$ we can apply the first recurrence (except finitely many points near the origin):

```
In[23]:= MyQRoots[lcfs[[1]] /. qn -> q, qj]
```

```
Out[23]= {0, 1, 2, 3, 4, 5, 6}
```

```
In[24]:= MyQRoots[lcfs[[1]] /. qn -> q^2, qj]
```

```
Out[24]= {1, 2, 3, 4}
```

```
In[26]:= MyQRoots[lcfs[[1]] /. qn -> q^3, qj]
```

```
Out[26]= {1, 2, 3, 4}
```

```
In[27]:= MyQRoots[lcfs[[1]] /. qn -> q^4, qj]
```

```
Out[27]= {1, 2, 3, 4, 5}
```

On the lines $j = 1, 2, 3, 4$ we can apply the last recurrence (except finitely many points near the origin):

```
In[28]:= MyQRoots[lcfs[[5]] /. qj -> q, qn]
```

```
Out[28]= {0, 1, 2, 3, 4, 5}
```

```
In[29]:= MyQRoots[lcfs[[5]] /. qj -> q^2, qn]
```

```
Out[29]= {0, 1, 2, 3, 4, 5}
```

```
In[30]:= MyQRoots[lcfs[[5]] /. qj -> q^3, qn]
```

```
Out[30]= {0, 1, 2, 3, 4, 5}
```

```
In[31]:= MyQRoots[lcfs[[5]] /. qj -> q^4, qn]
```

```
Out[31]= {0, 1, 2, 3, 4, 5, 6}
```

In the area $j > 4$ and $n > 4$, we can safely apply recurrence 3 (no singularities here):

```
In[33]:= exps = DeleteCases[Flatten[MapIndexed[(Exponent[#1, q] + {n, j}).(#2 - 1) &,
      CoefficientList[lcfs[[3]], {qn, qj}], {2}], -Infinity];
```

```
In[35]:= i1 = 1;
While[i1 ≤ Length[exps],
  i2 = 1;
  While[i2 ≤ Length[exps] && Simplify[exps[[i1]] < exps[[i2]], n ≥ 5 && j ≥ 5] != True, i2++];
  If[i2 > Length[exps], i1++, exps = Drop[exps, {i1}]];
]
```

```
In[37]:= exps
```

```
Out[37]= {14 + 24 j + 26 n, 16 + 23 j + 27 n, 7 + 22 j + 32 n,
  5 + 21 j + 33 n, 9 + 20 j + 34 n, 7 + 19 j + 35 n, 4 + 18 j + 36 n}
```

```
In[38]:= Table[
  {k, " ", Simplify[Reduce[Simplify[n ≥ 5 && j ≥ 5 && And@@(exps[[k]] > # & /@ Drop[exps, {k}])]],
  Element[{n, j}, Integers]], {k, Length[exps]}] // TableForm
Out[38]//TableForm=
  1   n ≥ 5 && 3 + j ≥ 3 n
  2   False
  3   n ≥ 5 && j ≥ 2 + n && 4 + j ≤ 3 n
  4   False
  5   j ≤ n && ((j ≥ 5 && n < 7 && n ≥ 5) || (n ≥ 7 && n ≤ 1 + j))
  6   False
  7   n > 8 && j ≥ 5 && 3 + j < n
```

The cases $j=n+1$, $j+2=n$, $j+3=n$ are not contained in the above regions and have to be checked separately:

```
In[41]:= MyQRroots[lcfs[[3]] /. qj → q * qn, qn]
```

```
Out[41]= {0, 1, 2, 3}
```

```
In[42]:= MyQRroots[lcfs[[3]] /. qn → q^2 * qj, qj]
```

```
Out[42]= {0, 1, 2}
```

```
In[43]:= MyQRroots[lcfs[[3]] /. qn → q^3 * qj, qj]
```

```
Out[43]= {0, 1, 2}
```

■ Prove correctness of the certificate

We have computed a creative telescoping relation for $q^{n+j-1} \left(\binom{n+j-2}{n-1}_q + q \binom{n+j-1}{n}_q \right) c(n, j)$. We convince ourselves that this operator indeed is in the corresponding annihilating ideal. We first load our solution and assemble the whole operator.

```
ansatz = << "ansatz3.m";
denom = << "denom3.m";
solution = << "solution3.m";
ctrel = ansatz /. myDen[a1__][a2_] → (denom[[a1]] /. qj → a2) /. solution;
```

This time, the reduction is more involved (more reduction steps have to be done). Our first attempt was interrupted by a computer crash, after one week, when it consumed about 33GB and was halfway through the reduction.

Therefore we thought of a different strategy that finally was successful. The idea is to avoid any summation that occurs during the reduction, by considering each term in a polynomial separately. Assume we start with the polynomial p ; we perform one reduction step on $LT(p)$, the leading term of p (and later also for each of the remaining terms). The result is in general again a polynomial p' consisting of several terms; we proceed as before by treating them all separately. Note that no addition of rational functions has to be done, only multiplications. Of course, this recursive branching yields an exponential number of terms in the end, and therefore is not a good idea in general, but in our example it is feasible: we obtain about 30 rational functions for each monomial under the stairs, which finally have to be added together. Since there are 10 monomials under the stairs, we have divided the problem into 10 smaller subproblems.

This strategy is implemented in the procedure `MyReduce`; it stores the intermediate results in the given directory.

```

MyReduce[poly_OrePolynomial, ann1_List, dir_String] :=
Module[{ann, lann, lpps, t, algebra, order, uts, fcount, todo, pos, red},
  uts = UnderTheStaircase[LeadingExponent /@ ann1];
  ann = SortBy[ann1, ByteCount];
  lann = Length[ann];
  lpps = LeadingExponent /@ ann;
  algebra = Together[OreAlgebra[ann]]; order = MonomialOrder[ann[[1]]];
  t = OrePolynomial[#, algebra, order] & /@ poly[[1]];
  fcount = 1; todo = {};

  (* store all terms of the input in order to treat them later. *)
  While[t != {}, Put[First[t], dir <> "myred" <> ToString[fcount] <> ".m"];
    If[Not[MemberQ[uts, LeadingExponent[First[t]]]], AppendTo[todo, fcount]];
    fcount++;
    t = Rest[t];
  ];

  While[todo != {},
    t = Get[dir <> "myred" <> ToString[First[todo]] <> ".m"];
    todo = Rest[todo];
    pos = 1;
    While[pos ≤ lann && Not[And@@ Thread[lpps[[pos]] ≤ LeadingExponent[t]], pos++];
    red = OreTimes[
      OrePolynomial[{1, LeadingExponent[t] - lpps[[pos]]}, algebra, order], ann[[pos]];
    t = Together[-LeadingCoefficient[t] / LeadingCoefficient[red]];
    t = OrePolynomial[{{Together[t * #[[1]], #[[2]]]}, algebra, order] & /@ Rest[First[red]]];
    While[t != {}, Put[First[t], dir <> "myred" <> ToString[fcount] <> ".m"];
      If[Not[MemberQ[uts, LeadingExponent[First[t]]]], AppendTo[todo, fcount]];
      fcount++;
      t = Rest[t];
    ];
  ];
];

```

The following command produces 360 files (altogether about 5GB).

```
MyReduce[ctrel, ann, ""]
```

Since the polynomials under consideration are partially factored, a tremendous speed-up can be achieved by using special purpose procedures for certain polynomial operations.

```

MyExponent[poly_, var_] :=
  If[Head[poly] != Times, Exponent[poly, var], Total[Exponent[#, var] & /@ (List@@poly)]];

```

```

(* returns {gcd(p11,p21), p11/gcd, p21/gcd}. *)
(* The strategy is to compute the gcd pairwise for all factors. *)
MyPolynomialGCD[p11_, p21_] :=
Module[{a1, a2, p1 = p11, p2 = p21, g, i, j, ex1, ex2, gcd, todo, vars},
  vars = Variables[{p1, p2}];
  p1 = If[Head[p1] === Times, List @@ p1, {p1}];
  p2 = If[Head[p2] === Times, List @@ p2, {p2}];
  gcd = Intersection[p1, p2];
  p1 = Complement[p1, gcd];
  p2 = Complement[p2, gcd];
  gcd = Times @@ gcd;
  ex1 = Total[MyExponent[#, vars]] & /@ p1;
  ex2 = Total[MyExponent[#, vars]] & /@ p2;
  todo = SortBy[
    Flatten[Table[{i, j, ex1[[i]] + ex2[[j]]}, {i, Length[p1]}, {j, Length[p2]}], 1], Last];
  While[todo != {},
    {i, j} = Take[First[todo], 2];
    todo = Rest[todo];
    g = PolynomialGCD[p1[[i]], p2[[j]]];
    If[g != 1, gcd *= g;
      p1[[i]] = Together[p1[[i]] / g]; p2[[j]] = Together[p2[[j]] / g];
      If[p1[[i]] === 1, todo = DeleteCases[todo, {i, _, _}]];
      If[p2[[j]] === 1, todo = DeleteCases[todo, {_, j, _}]];
      ex1 = Total[MyExponent[g, vars]];
      todo =
        SortBy[todo /. {i, a1_, a2_} => {i, a1, a2 - ex1} /. {a1_, j, a2_} => {a1, j, a2 - ex1}, Last];
    ];
  ];
  Return[{gcd, Times @@ p1, Times @@ p2}];
];

(* takes a list of rational functions, and computes their sum in togethered form. *)
(* In the option Replace a list of
substitutions can be given (for modular computations). *)
Options[TogetherTotal] = {Replace -> {{}}, NormalizeVector -> True};
TogetherTotal[p11_List, opts : ((_Rule | _RuleDelayed) ...)] :=
Module[{i, pl = p11, replace, normalize, repl, pl0, polymod,
  pli, vars, subs, plmod, pairs, gcd, p1, p2, max, t, prime, result},
  {replace, normalize} = {Replace, NormalizeVector} /. {opts} /. Options[TogetherTotal];

  (* if not otherwise stated,
we multiply by the common denominator, and remove content. *)
  If[normalize === True,
    gcd = PolynomialLCM @@ Denominator[pl];
    If[gcd != 1, pl = Together[gcd * pl]];
    gcd = Fold[PolynomialGCD, First[pl], Rest[pl]] & [SortBy[pl, ByteCount]];
    If[gcd != 1, pl = Together[pl / gcd]];
  ];

  result = {};
  pl0 = pl;
  Do[

```



```

pl = p10 /. replace[[repl]];

(* By modular computations we try to find out which pairs
of polynomials have the largest gcd (those will be treated first). *)
vars = Variables[pl];
subs = Table[NextPrime[19, 3 * i], {i, Length[vars] - 1}];
prime = NextPrime[Developer`$MaxMachineInteger, -1];
polymod = Function[poly, If[Head[poly] === Times,
  PolynomialMod[#, prime] & /@ poly, PolynomialMod[poly, prime]]];
plmod = polymod[pl /. Thread[Complement[vars, {#}] -> subs]] & /@ vars;

(* List of indices which entries in pl are valid. *)
pli = Range[Length[pl]];

pairs = Function[pair, t = Sum[Exponent[PolynomialGCD@@
  Append[plmod[[i, pair]], Modulus -> prime], vars[[i]], {i, Length[vars]}];
  {pair, t}] /@ Subsets[pli, {2}];

(* Now we add the polynomials pairwise: first remove the gcd,
add the rest, and multiply again by the gcd. *)
While[pairs != {},
  max = Max[#[[2]] & /@ pairs];
  {p1, p2} = First[pairs[[Position[pairs, {_, max}][[1, 1]]]];
  gcd = MyPolynomialGCD[p1[[p1]], p1[[p2]]];
  pl[[p1]] = First[gcd] * Expand[Total[Rest[gcd]]];

  (* Update modular data and list of pairs and indices. *)
  Do[plmod[[i, p1]] = polymod[p1[[p1]] /. Thread[Complement[vars, {vars[[i]]}] -> subs]];
  {i, Length[vars]}];
  pairs = DeleteCases[pairs, {{p1, _} | {_, p1} | {p2, _} | {_, p2}, _];
  pli = Complement[pli, {p2}];

  If[pl[[p1]] === 0,
    pli = Complement[pli, {p1}];
    , pairs = Join[pairs, Function[pair, t = Sum[Exponent[PolynomialGCD@@ Append[
      plmod[[i, pair]], Modulus -> prime], vars[[i]], {i, Length[vars]}];
      {pair, t}] /@ ({p1, #} & /@ Complement[pli, {p1}])];
  ];
];
AppendTo[result, Total[pl[[pli]]]];

, {repl, Length[replace]}; (* end of Do *)

Clear[pl, p10, subs];
Return[result];
];

```

Now we are equipped with everything that is needed. We demonstrate for one monomial (viz. $QS[qn, q^n]^3$) how to proceed. The remaining ones can be treated in the same fashion, although they take longer. For the bigger cases the modular approach is highly preferable.

```

rats = {};
Do[
  t = Get["myred" <> ToString[nn] <> ".m"];
  If[LeadingExponent[t] === {3, 0}, AppendTo[rats, LeadingCoefficient[t]]];
  , {nn, 360}];

Length[rats]

30

(* takes about 6 hours. *)
TogetherTotal[rats]

{0}

```



```
TraditionalForm[HoldForm[Sum[a[n, j] c[n, j], {j, 1, n}] =
  Sum[q^(n + j - 1) * (QBinomial[n + j - 2, n - 1, q] + q * QBinomial[n + j - 1, n, q]) * c[n, j],
    {j, 1, n}] + (1 + q^n) - c[n, n - 1]]]
```

$$\sum_{j=1}^n a(n, j) c(n, j) = \sum_{j=1}^n q^{n+j-1} \left(\binom{n+j-2}{n-1}_q + q \binom{n+j-1}{n}_q \right) c(n, j) + (1 + q^n) - c(n, n-1)$$

```
(* This computation takes 108h and needs about 3GB of memory. *)
```

```
nm = $NormalizeMethod; $NormalizeMethod = "gcd";
recPlus = DFinitePlus[{Expand[pp]}, Annihilator[1 + q^n, QS[qn, q^n]], recDiag1];
$NormalizeMethod = nm;
```

```
ByteCount[recPlus]
```

```
420 829 424
```

```
Support[recPlus]
```

```
{{Sqn,q^12, Sqn,q^11, Sqn,q^10, Sqn,q^9, Sqn,q^8, Sqn,q^7, Sqn,q^6, Sqn,q^5, Sqn,q^4, Sqn,q^3, Sqn,q^2, Sqn,q, 1}}
```

Now, after we have computed a recurrence for the left-hand side, we care about the right-hand side.

```
b[n_Integer] :=
  Product[((1 - q^(i + j + k - 1)) / (1 - q^(i + j + k - 2)))^2, {i, n}, {j, i, n}, {k, j, n}]
```

By simple hand calculations we found:

```
b[n] / b[n - 1] == QPochhammer[q^(2 n), q, n]^2 / QPochhammer[q^n, q^2, n]^2
```

$$\frac{b[n]}{b[-1 + n]} = \frac{\text{QPochhammer}[q^{2n}, q, n]^2}{\text{QPochhammer}[q^n, q^2, n]^2}$$

```
FunctionExpand[Table[%, {n, 10}]]
```

```
{True, True, True, True, True, True, True, True, True, True}
```

Hence we get the following second-order recurrence for the right-hand side:

```
annb = Together[Factor[
  Annihilator[QPochhammer[q^(2 n), q, n]^2 / QPochhammer[q^n, q^2, n]^2, QS[qn, q^n]]]]
  {(1 + qn)^2 (1 + q qn)^2 (-1 + q qn^2)^2 (-1 + q^3 qn^2)^2 Sqn,q^2 - (1 + q qn + q^2 qn^2)^2 (-1 + q qn^3)^2 (-1 + q^5 qn^3)^2}
```

The following computation proves that recPlus is a left multiple of the recurrence for b[n]/b[n-1]:

```
Timing[reduce = OreReduce[First[recPlus], annb];]
```

```
{2521.63, Null}
```

```
reduce
```

```
0
```

We investigate the leading coefficient of the recurrence recPlus:

```
lcoeff = LeadingCoefficient[recPlus];
```

```
Select[First /@ FactorList[lcoeff], Exponent[#, qn] == 1 &]
```

```
{1 + q^7 qn, 1 + q^8 qn, 1 + q^9 qn, 1 + q^10 qn, 1 + q^11 qn}
```

→ The recurrence has no singularities for positive n . Alternatively, this fact can be shown as follows (avoiding the costly factorization). Considering $lcoeff$ as a polynomial in q^n , we find that the q -degree of the leading coefficient is maximal among the q -degrees of all coefficients. This proves that for positive n , the polynomial cannot become zero.

```
Coefficient[lcoeff, qn ^ Exponent[lcoeff, qn]]
```

$$q^{1554} + 2 q^{1555} - q^{1556} - 19 q^{1557} - 65 q^{1558} - 143 q^{1559} - 239 q^{1560} - 316 q^{1561} - 331 q^{1562} - 256 q^{1563} - 95 q^{1564} +$$

$$112 q^{1565} + 310 q^{1566} + 451 q^{1567} + 511 q^{1568} + 496 q^{1569} + 422 q^{1570} + 308 q^{1571} + 166 q^{1572} + 9 q^{1573} -$$

$$137 q^{1574} - 243 q^{1575} - 284 q^{1576} - 260 q^{1577} - 194 q^{1578} - 118 q^{1579} - 58 q^{1580} - 22 q^{1581} - 6 q^{1582} - q^{1583}$$

```
Max[Exponent[#, q] & /@ CoefficientList[lcoeff, qn]]
```

```
1583
```

Hence we have to compare 12 initial values for $n=1, \dots, 12$:

```
Table[Together[Sum[a[n, j] * c[n, j], {j, n}] - b[n] / b[n - 1]], {n, 12}]
```

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

This completes the proof of (Okada). *Quod erat demonstrandum.*