The Gerrymandering Problem

OEIS A348456: Number of ways to dissect a 2*n X 2*n chessboard into two polyominoes each of area 2*n^2.

Code

In[49]:=

```
SetDirectory[NotebookDirectory[]];
Quiet[<< Combinatorica`];

(* Determine chunks (runs) of same elements in a {0,1}-vector. *)
Chunks[vec_] :=
  Map[Last, SplitBy[Transpose[{vec, Range[Length[vec]]}], First], {2}];
(* Construct all non-crossing arc configurations
  on n nodes with alternating colors. *)
NoncrossingConnections[n_Integer] := NoncrossingConnections[n] =
   Select[Tuples[SetPartitions[Table[i, {i, #, n, 2}]] & /@ {1, 2}],
    Cases[Subsets[Join @@ #, {2}], {{___, a1_, ___, a2_, ___}, {___, b1_, ___,
          b2_, ___}} /; (a1 < b1 < a2 < b2 || b1 < a1 < b2 < a2)] === {} &];
NoncrossingConnections[chunks_List] := Map[Flatten,
   NoncrossingConnections[Length[chunks]] /. a_Integer :> chunks[[a]], {3}];

(* Rows and columns of the transfer matrix
 are labeled by states of the form {col, co0, co1}. *)
(* Example: {col, co0, co1} =
 {{1,0,0,1,0,1,1,0}, {{2,3},{5,8}}, {{1},{4},{6,7}}} means
  that the 0's at positions 5 and 8 are connected,
while the three chunks of 1's are not. *)
(* Step (1): Construct the set of all possible (reachable) states. *)
MatrixLabels1[n_] := Join[Flatten[
    Function[v, If[v[[1]] === 0, {v, #1, #2}, {v, #2, #1}] & @@@
       NoncrossingConnections[Chunks[v]]] /@ Tuples[{0, 1}, 2 n], 1],
   (* The following two extra cases represent situations where a
    unicolor column has fields of the other color before it,
   hence it can only be followed by the same column. *)
   {{Table[0, {2 n}], {}, {}}, {Table[1, {2 n}], {}, {}}}];
(* Step (2): Discard all uninteresting states. *)
MatrixLabels[n_] := MatrixLabels[n] =
   DeleteCases[MatrixLabels1[n],
    {_, {___, c0a_, ___, c0b_, ___}, {___, c1a_, ___, c1b_, ___}} /;
     (Max[c0a] + 1 == Min[c1a] && Max[c1a] == Min[c0b] - 1 && Max[c0b] < Max[c1b]) ||
      (Max[c0a] + 1 == Min[c1b] && Max[c1b] == Min[c0b] - 1 && Min[c1a] < Min[c0a]) ||
```

```
      (Max[c1a] + 1 == Min[c0a] && Max[c0a] == Min[c1b] - 1 && Max[c1b] < Max[c0b]) ||
      (Max[c1a] + 1 == Min[c0b] && Max[c0b] == Min[c1b] - 1 && Min[c0a] < Min[c1a])];

(* Determine which chunks of the new column are connected,
based on the connection information of the preceding column. *)
(* Examples:
 ComposeConnections[{{1,3},{5,7},{9}},{{1},{3,4,5},{7,8},{10}}];
 ComposeConnections[{{1,3,9},{5},{7}},{{1,2},{5,6},{8,9}}];
 ComposeConnections[{{3,4}},{{1,2},{5,6}}]; *)
ComposeConnections[co_, ch1_] :=
  Module[{ch = ch1, s, s1, conn = {}},
   While[ch =!= {},
    AppendTo[conn,
     FixedPoint[Function[s,
       s1 = Flatten[Select[co, Intersection[#, s] =!= {} &]];
       Union[s, Flatten[Select[ch, Intersection[#, s1] =!= {} &]]]],
      First[ch]]];
    ch = Select[ch, Intersection[#, Last[conn]] === {} &];
   ];
   Return[conn];
  ];

(* Returns a sparse square {0,1}-matrix of dimension Length[MatrixLabels]. *)
(* The true transfer matrix
 (containing powers of x) is XVector[n]*BuildMatrix[n].
  Note that each row contains only entries 0 or x^k with the same k. *)
BuildMatrix[n_] :=
  Module[{ml = MatrixLabels[n], dim, chs,
    mlspl, pos, mat, vecs = Tuples[{0, 1}, 2 n], row, nl, p1},
   dim = Length[ml];
   chs = Function[v,
      Take[Chunks[v], {#, -1, 2}] & /@ If[v[[1]] === 0, {1, 2}, {2, 1}]] /@ vecs;
   mlspl = Append[SplitBy[Drop[ml, -3], First], Take[ml, -3]];
   pos = Most[FoldList[Plus, 0, Length /@ mlspl]];
   mat = SparseArray[{}, {1, dim}];
   Function[{vv, c0, c1},
     row = SparseArray[{}, {dim}];
     MapThread[Function[{v, ch, mlv, p},
       If[(Or @@ (MatchQ[v[[#]], {(1) ..}] & /@ c0)) ||
          (Or @@ (MatchQ[v[[#]], {(0) ..}] & /@ c1)),
         Which[
          MatchQ[v, {(0) ..}] && Length[c1] ≤ 1, row[[-2]] += 1,
          MatchQ[v, {(1) ..}] && Length[c0] ≤ 1, row[[-1]] += 1]
```

```
          ,
          If[c0 === c1 === {},
            Which[
             vv === v === Table[0, {2 n}], row[[-2]] += 1,
             vv === v === Table[1, {2 n}], row[[-1]] += 1]
            ,
            nl = Prepend[MapThread[ComposeConnections, {{c0, c1}, ch}], v];
            If[(p1 = FirstPosition[mlv, nl]) =!=
              Missing["NotFound"], row[[p + p1[[1]]]] += 1];
           ];
         ];
       ], {vecs, chs, mlspl, pos}];
      AppendTo[mat, row];
     ] @@@ ml;
    Return[Rest[mat]];
   ];
XVector[n_] := (x^Count[#[[1]], 0]) & /@ MatrixLabels[n];
(* Encodes possible start configurations
 (i.e. those with no nontrivial connections). *)
StartVector[n_] :=
  If[Length[Join[#2, #3]] === Length[Chunks[#1]], x^Count[#1, 0], 0] &@@@
   MatrixLabels[n];
(* {0,1}-vector that picks only those configurations
  that are fully connected. *)
EndVector[n_] := If[MatchQ[#2, {} | {_}] && MatchQ[#3, {} | {_}], 1, 0] &@@@
   MatrixLabels[n];
```

$n = 1$

In[17]:=
```
MatrixForm[BuildMatrix[1].DiagonalMatrix[XVector[1]]]
```

Out[17]//MatrixForm=

$$\begin{pmatrix} x^2 & x & x & 0 & 0 & 1 \\ 0 & x & 0 & 0 & x^2 & 1 \\ 0 & 0 & x & 0 & x^2 & 1 \\ 0 & x & x & 1 & x^2 & 0 \\ 0 & 0 & 0 & 0 & x^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In[18]:=
```
StartVector[1]
```

Out[18]= $\{x^2, x, x, 1, 0, 0\}$

In[20]:=
```
EndVector[1]
```

Out[20]= {1, 1, 1, 1, 1, 1}

In[21]:=
```
Expand[StartVector[1].BuildMatrix[1] * XVector[1]]
```
Out[21]=
$$\{x^4,\ x + x^2 + x^3,\ x + x^2 + x^3,\ 1,\ x^2 + 2\,x^3,\ 2\,x + x^2\}$$

In[22]:=
```
%.EndVector[1]
```
Out[22]=
$$1 + 4\,x + 4\,x^2 + 4\,x^3 + x^4$$

$n = 2$

In[27]:=
```
Print /@ MatrixLabels[2];
```
```
{{0, 0, 0, 0}, {{1, 2, 3, 4}}, {}}
{{0, 0, 0, 1}, {{1, 2, 3}}, {{4}}}
{{0, 0, 1, 0}, {{1, 2, 4}}, {{3}}}
{{0, 0, 1, 0}, {{1, 2}, {4}}, {{3}}}
{{0, 0, 1, 1}, {{1, 2}}, {{3, 4}}}
{{0, 1, 0, 0}, {{1, 3, 4}}, {{2}}}
{{0, 1, 0, 0}, {{1}, {3, 4}}, {{2}}}
{{0, 1, 0, 1}, {{1, 3}}, {{2}, {4}}}
{{0, 1, 0, 1}, {{1}, {3}}, {{2, 4}}}
{{0, 1, 1, 0}, {{1, 4}}, {{2, 3}}}
{{0, 1, 1, 0}, {{1}, {4}}, {{2, 3}}}
{{0, 1, 1, 1}, {{1}}, {{2, 3, 4}}}
{{1, 0, 0, 0}, {{2, 3, 4}}, {{1}}}
{{1, 0, 0, 1}, {{2, 3}}, {{1, 4}}}
{{1, 0, 0, 1}, {{2, 3}}, {{1}, {4}}}
{{1, 0, 1, 0}, {{2}, {4}}, {{1, 3}}}
{{1, 0, 1, 0}, {{2, 4}}, {{1}, {3}}}
{{1, 0, 1, 1}, {{2}}, {{1, 3, 4}}}
{{1, 0, 1, 1}, {{2}}, {{1}, {3, 4}}}
{{1, 1, 0, 0}, {{3, 4}}, {{1, 2}}}
{{1, 1, 0, 1}, {{3}}, {{1, 2, 4}}}
{{1, 1, 0, 1}, {{3}}, {{1, 2}, {4}}}
{{1, 1, 1, 0}, {{4}}, {{1, 2, 3}}}
{{1, 1, 1, 1}, {}, {{1, 2, 3, 4}}}
{{0, 0, 0, 0}, {}, {}}
{{1, 1, 1, 1}, {}, {}}
```

In[23]:=
```
MatrixForm[mat2 = BuildMatrix[2]]
```

Out[23]//MatrixForm=

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 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 & 1 & 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 & 1
\end{pmatrix}$$

In[25]:=
```
StartVector[2]
```

Out[25]= $\{x^4, x^3, 0, x^3, x^2, 0, x^3, 0, 0, 0, x^2, x, x^3, 0, x^2, 0, 0, 0, x, x^2, 0, x, x, 1, 0, 0\}$

In[26]:=
```
EndVector[2]
```

Out[26]= {1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1}

In[24]:=
```
Nest[Expand[(#.mat2) * XVector[2]] &, StartVector[2], 3].EndVector[2]
```

Out[24]= $1 + 16\,x + 24\,x^2 + 48\,x^3 + 85\,x^4 + 116\,x^5 + 132\,x^6 + 136\,x^7 + 140\,x^8 + 136\,x^9 + 132\,x^{10} + 116\,x^{11} + 85\,x^{12} + 48\,x^{13} + 24\,x^{14} + 16\,x^{15} + x^{16}$

In[28]:=
```
(* Verify this number by a naive, brute-force computation *)
MyConnectedQ[set_] := FixedPoint[Function[s,
      Intersection[Join @@ ({#, # + {1, 0}, # + {-1, 0}, # + {0, 1}, # + {0, -1}} & /@ s),
       set]], {First[set]}] === Sort[set];
With[{t = Tuples[Range[4], 2]},
 test = Select[Subsets[t, {8}], MyConnectedQ[#] && MyConnectedQ[Complement[t, #]] &];
 Length[test]
]
```

Out[29]= 140

$n = 3$

In[30]:=
```
Length[MatrixLabels[3]]
```
Out[30]= 154

In[31]:=
```
Timing[mat3 = BuildMatrix[3];]
```
Out[31]= {0.366871, Null}

In[33]:=
```
svec3 = StartVector[3];
evec3 = EndVector[3];
xvec3 = XVector[3];
mat3x = mat3.SparseArray[DiagonalMatrix[xvec3]];
```

In[37]:=
```
(* Most stupid way: powering the matrix, then multiply with vector *)
Timing[test1 = Expand[svec3.MatrixPower[mat3x, 5].evec3];]
```
Out[37]= {6.64097, Null}

In[38]:=
```
(* Better: do only matrix-vector multiplications *)
Timing[test2 = Nest[Expand[#.mat3x] &, svec3, 5].evec3;]
```
Out[38]= {0.261441, Null}

In[39]:=
```
(* Even better: use {0,1} matrix and x-vector *)
Timing[test3 = Nest[Expand[(#.mat3) * xvec3] &, svec3, 5].evec3;]
```
Out[39]= {0.102886, Null}

In[41]:=
```
(* Using PolynomialMod *)
Timing[Nest[PolynomialMod[(#.mat3) * xvec3, x^19] &, svec3, 5].evec3]
```
Out[41]= $\{0.121917,\ 1 + 36\,x + 60\,x^2 + 144\,x^3 + 357\,x^4 + 904\,x^5 + 2292\,x^6 + 5616\,x^7 + 12\,850\,x^8 + 26\,648\,x^9 + 48\,898\,x^{10} + 78\,224\,x^{11} + 108\,809\,x^{12} + 133\,300\,x^{13} + 148\,574\,x^{14} + 156\,304\,x^{15} + 159\,509\,x^{16} + 160\,700\,x^{17} + 161\,036\,x^{18}\}$

In[•]:=
```
(* Using evaluation-interpolation *)
Timing[test4 = Expand[InterpolatingPolynomial[Table[xvec0 = xvec3 /. x → x0;
      Nest[((#.mat3) * xvec0) &, svec3 /. x → x0, 5].evec3, {x0, 40}], x]]]
```
Out[•]= $\{0.05981,\ 1 + 36\,x + 60\,x^2 + 144\,x^3 + 357\,x^4 + 904\,x^5 + 2292\,x^6 + 5616\,x^7 + 12\,850\,x^8 + 26\,648\,x^9 + 48\,898\,x^{10} + 78\,224\,x^{11} + 108\,809\,x^{12} + 133\,300\,x^{13} + 148\,574\,x^{14} + 156\,304\,x^{15} + 159\,509\,x^{16} + 160\,700\,x^{17} + 161\,036\,x^{18} + 160\,700\,x^{19} + 159\,509\,x^{20} + 156\,304\,x^{21} + 148\,574\,x^{22} + 133\,300\,x^{23} + 108\,809\,x^{24} + 78\,224\,x^{25} + 48\,898\,x^{26} + 26\,648\,x^{27} + 12\,850\,x^{28} + 5616\,x^{29} + 2292\,x^{30} + 904\,x^{31} + 357\,x^{32} + 144\,x^{33} + 60\,x^{34} + 36\,x^{35} + x^{36}\}$

In[43]:=
```
test1 === test2 === test3 === test4
```
Out[43]= True

In[44]:=
```
(* How much memory is saved by using SparseArray and 0/1's. *)
ByteCount /@ {mat3, mat3x, Normal[mat3], Normal[mat3x]}
```

Out[44]= {83 560, 205 096, 579 096, 700 632}

$n = 4$

In[45]:=
```
(* Size of the transfer matrix if uninteresting states are not discarded. *)
Timing[Length[MatrixLabels1[4]]]
```

Out[45]= {0.047372, 1836}

In[47]:=
```
(* Improved version: discarding uninteresting states yields a smaller matrix. *)
Timing[Length[MatrixLabels[4]]]
```

Out[48]= {0.0615836, 1026}

In[•]:=
```
Timing[mat4 = BuildMatrix[4];]
```

Out[•]= {9.64768, Null}

In[•]:=
```
Timing[xvec4 = XVector[4]; svec4 = StartVector[4]; evec4 = EndVector[4];]
```

Out[•]= {0.028262, Null}

In[•]:=
```
Timing[test1 = Nest[Expand[(#.mat4) * xvec4] &, svec4, 7].evec4;]
```

Out[•]= {2.06528, Null}

```
(* Comparison to using a normal matrix (not SparseArray). *)
mat4a = Normal[mat4];
Print[ByteCount /@ {mat4, mat4a}];
Timing[test2 = Nest[Expand[(#.mat4a) * xvec4] &, svec4, 7].evec4;]
```

{1 418 160, 26 315 864}

Out[•]= {10.6691, Null}

```
(* Putting the x-powers into the matrix (not using SparseArray). *)
mat4a = Normal[mat4].DiagonalMatrix[xvec4];
Print[ByteCount /@ {mat4, mat4a}];
Timing[test3 = Nest[Expand[#.mat4a] &, svec4, 7].evec4;]
```

{1 418 160, 28 674 264}

Out[•]= {29.7066, Null}

```
(* Using sparse matrix and (palindromic) evaluation-interpolation,
modulo a single, sufficiently large prime. *)
Timing[
 pr = NextPrime[2^41, -1];
 vals = Table[xvmod = Mod[xvec4 /. x → xx, pr];
   {xx, Nest[Mod[(#.mat4) * xvmod, pr] &, svec4 /. x → xx, 7].evec4}, {xx, 2, 34}];
 vals = PolynomialMod[Join[vals, {1 / #1, #2 / #1^64} &@@@ vals], pr];
 test4 = PolynomialMod[InterpolatingPolynomial[vals, x, Modulus → pr], pr];
]
```

Out[•]= {0.274803, Null}

```
(* Sparsity: 96% of the matrix entries are zeros. *)
N[Total[Count[#, 0] & /@ mat4a] / (Times @@ Dimensions[mat4a])]
```

Out[•]= 0.958194

In[•]:= `test1 === test2 === test3 === test4`

Out[•]= True

In[•]:= `test1`

Out[•]= $1 + 64\,x + 112\,x^2 + 288\,x^3 + 777\,x^4 + 2180\,x^5 + 6292\,x^6 + 18\,424\,x^7 + 54\,250\,x^8 + 159\,464\,x^9 +$
$463\,922\,x^{10} + 1\,323\,540\,x^{11} + 3\,670\,927\,x^{12} + 9\,821\,592\,x^{13} + 25\,174\,054\,x^{14} + 61\,431\,424\,x^{15} +$
$141\,910\,433\,x^{16} + 308\,667\,036\,x^{17} + 628\,870\,790\,x^{18} + 1\,194\,000\,508\,x^{19} + 2\,102\,168\,237\,x^{20} +$
$3\,417\,076\,552\,x^{21} + 5\,113\,453\,522\,x^{22} + 7\,043\,986\,348\,x^{23} + 8\,969\,982\,764\,x^{24} +$
$10\,657\,077\,648\,x^{25} + 11\,971\,286\,872\,x^{26} + 12\,902\,859\,116\,x^{27} + 13\,520\,045\,532\,x^{28} +$
$13\,910\,495\,776\,x^{29} + 14\,146\,489\,836\,x^{30} + 14\,273\,772\,524\,x^{31} + 14\,314\,228\,378\,x^{32} +$
$14\,273\,772\,524\,x^{33} + 14\,146\,489\,836\,x^{34} + 13\,910\,495\,776\,x^{35} + 13\,520\,045\,532\,x^{36} +$
$12\,902\,859\,116\,x^{37} + 11\,971\,286\,872\,x^{38} + 10\,657\,077\,648\,x^{39} + 8\,969\,982\,764\,x^{40} +$
$7\,043\,986\,348\,x^{41} + 5\,113\,453\,522\,x^{42} + 3\,417\,076\,552\,x^{43} + 2\,102\,168\,237\,x^{44} +$
$1\,194\,000\,508\,x^{45} + 628\,870\,790\,x^{46} + 308\,667\,036\,x^{47} + 141\,910\,433\,x^{48} + 61\,431\,424\,x^{49} +$
$25\,174\,054\,x^{50} + 9\,821\,592\,x^{51} + 3\,670\,927\,x^{52} + 1\,323\,540\,x^{53} + 463\,922\,x^{54} + 159\,464\,x^{55} +$
$54\,250\,x^{56} + 18\,424\,x^{57} + 6292\,x^{58} + 2180\,x^{59} + 777\,x^{60} + 288\,x^{61} + 112\,x^{62} + 64\,x^{63} + x^{64}$

$n = 5$

```
(* Size of the transfer matrix if uninteresting states are not discarded. *)
Timing[Length[MatrixLabels1[5]]]
```

Out[•]= {0.221462, 17 578}

```
(* Size of the transfer matrix after discarding uninteresting states. *)
Timing[Length[MatrixLabels[5]]]
```

Out[•]= {0.578678, 7222}

```
In[•]:= Timing[mat5 = BuildMatrix[5];]
Out[•]= {306.785, Null}

        (* Sparsity: 98.4% of the matrix entries are zeros. *)
        N[Count[Normal[mat5], 0, {2}] / (Times @@ Dimensions[mat5])]
Out[•]= 0.98404

In[•]:= Timing[xvec5 = XVector[5]; svec5 = StartVector[5]; evec5 = EndVector[5];]
Out[•]= {0.196329, Null}

In[•]:= Timing[res5 = Nest[Expand[(#.mat5) * xvec5] &, svec5, 9].evec5;]
Out[•]= {119.009, Null}

        (* Interesting: this computation takes half the time,
        although it is basically the same! *)
        mat5a = Transpose[mat5];
        Timing[res5a = Nest[Expand[(mat5a.#) * xvec5] &, svec5, 9].evec5;]
Out[•]= {56.1167, Null}
```

In[•]:=
```
res5
```

Out[•]=
$$1 + 100\,x + 180\,x^2 + 480\,x^3 + 1349\,x^4 + 3960\,x^5 + 12\,012\,x^6 + 37\,176\,x^7 + 116\,570\,x^8 + 368\,872\,x^9 + 1\,174\,058\,x^{10} + 3\,747\,348\,x^{11} + 11\,959\,935\,x^{12} + 38\,052\,792\,x^{13} + 120\,314\,294\,x^{14} + 376\,837\,672\,x^{15} + 1\,165\,797\,439\,x^{16} + 3\,552\,850\,964\,x^{17} + 10\,641\,225\,230\,x^{18} + 31\,256\,397\,052\,x^{19} + 89\,858\,259\,377\,x^{20} + 252\,365\,407\,912\,x^{21} + 691\,123\,921\,388\,x^{22} + 1\,842\,216\,561\,360\,x^{23} + 4\,770\,623\,156\,538\,x^{24} + 11\,979\,000\,435\,024\,x^{25} + 29\,106\,639\,791\,902\,x^{26} + 68\,287\,671\,300\,500\,x^{27} + 154\,329\,639\,366\,718\,x^{28} + 335\,128\,134\,321\,440\,x^{29} + 697\,345\,966\,308\,712\,x^{30} + 1\,386\,509\,498\,373\,508\,x^{31} + 2\,626\,434\,846\,621\,049\,x^{32} + 4\,726\,396\,963\,178\,680\,x^{33} + 8\,058\,420\,647\,101\,132\,x^{34} + 12\,987\,878\,424\,779\,208\,x^{35} + 19\,755\,965\,460\,931\,931\,x^{36} + 28\,344\,019\,886\,765\,832\,x^{37} + 38\,382\,657\,658\,335\,306\,x^{38} + 49\,173\,686\,293\,434\,612\,x^{39} + 59\,847\,092\,485\,693\,681\,x^{40} + 69\,595\,785\,198\,856\,128\,x^{41} + 77\,876\,412\,031\,730\,680\,x^{42} + 84\,486\,338\,430\,229\,472\,x^{43} + 89\,508\,636\,423\,456\,728\,x^{44} + 93\,187\,757\,293\,247\,704\,x^{45} + 95\,809\,331\,066\,889\,618\,x^{46} + 97\,623\,019\,871\,453\,288\,x^{47} + 98\,810\,656\,166\,647\,352\,x^{48} + 99\,485\,193\,719\,743\,672\,x^{49} + 99\,704\,315\,229\,167\,288\,x^{50} + 99\,485\,193\,719\,743\,672\,x^{51} + 98\,810\,656\,166\,647\,352\,x^{52} + 97\,623\,019\,871\,453\,288\,x^{53} + 95\,809\,331\,066\,889\,618\,x^{54} + 93\,187\,757\,293\,247\,704\,x^{55} + 89\,508\,636\,423\,456\,728\,x^{56} + 84\,486\,338\,430\,229\,472\,x^{57} + 77\,876\,412\,031\,730\,680\,x^{58} + 69\,595\,785\,198\,856\,128\,x^{59} + 59\,847\,092\,485\,693\,681\,x^{60} + 49\,173\,686\,293\,434\,612\,x^{61} + 38\,382\,657\,658\,335\,306\,x^{62} + 28\,344\,019\,886\,765\,832\,x^{63} + 19\,755\,965\,460\,931\,931\,x^{64} + 12\,987\,878\,424\,779\,208\,x^{65} + 8\,058\,420\,647\,101\,132\,x^{66} + 4\,726\,396\,963\,178\,680\,x^{67} + 2\,626\,434\,846\,621\,049\,x^{68} + 1\,386\,509\,498\,373\,508\,x^{69} + 697\,345\,966\,308\,712\,x^{70} + 335\,128\,134\,321\,440\,x^{71} + 154\,329\,639\,366\,718\,x^{72} + 68\,287\,671\,300\,500\,x^{73} + 29\,106\,639\,791\,902\,x^{74} + 11\,979\,000\,435\,024\,x^{75} + 4\,770\,623\,156\,538\,x^{76} + 1\,842\,216\,561\,360\,x^{77} + 691\,123\,921\,388\,x^{78} + 252\,365\,407\,912\,x^{79} + 89\,858\,259\,377\,x^{80} + 31\,256\,397\,052\,x^{81} + 10\,641\,225\,230\,x^{82} + 3\,552\,850\,964\,x^{83} + 1\,165\,797\,439\,x^{84} + 376\,837\,672\,x^{85} + 120\,314\,294\,x^{86} + 38\,052\,792\,x^{87} + 11\,959\,935\,x^{88} + 3\,747\,348\,x^{89} + 1\,174\,058\,x^{90} + 368\,872\,x^{91} + 116\,570\,x^{92} + 37\,176\,x^{93} + 12\,012\,x^{94} + 3960\,x^{95} + 1349\,x^{96} + 480\,x^{97} + 180\,x^{98} + 100\,x^{99} + x^{100}$$

$n = 6$

```
(* Size of the transfer matrix if uninteresting states are not discarded. *)
(* The corresponding BuildMatrix took 42h on radon1 (single processor). *)
Timing[Length[MatrixLabels1[6]]]
```

Out[•]=
```
{1.96027, 177 056}
```

```
(* Size of the transfer matrix after discarding uninteresting states. *)
Timing[Length[MatrixLabels[6]]]
```

Out[•]=
```
{7.42976, 52 650}
```

BuildMatrix[6] took about 20min on compute-grantley using 15 parallel processes.
For parallelization, we divided the matrix into chunks of 1000 rows.
Matrix (SparseArray) has ByteCount 264,509,768 and 195M file size.

```
(* Assemble the whole matrix from the parallelized chunks. *)
dim = Length[MatrixLabels[6]];
mat6 =
  Transpose[Join @@ Table[Get["mat6_" <> ToString[p] <> ".m"], {p, 1, dim, 1000}]];

xvec = XVector[6];
(* took 2956s on compute-grantley *)
Timing[res6 = Nest[Expand[xvec * (mat6.#)] &, StartVector[6], 11].EndVector[6];]

(* took 2651s on compute-grantley *)
Timing[res6 =
   Nest[PolynomialMod[xvec * (mat6.#), x^73] &, StartVector[6], 11].EndVector[6];]
```

With evaluation-interpolation and chinese remaindering, it took 20 minutes (without exploiting palindromicity) on compute-grantley, using 3 primes.

```
In[•]:= res6 = Get["res6.m"]
```

$$
\begin{aligned}
\text{Out}[\bullet]= \; & 1 + 144\,x + 264\,x^2 + 720\,x^3 + 2073\,x^4 + 6244\,x^5 + 19\,460\,x^6 + 61\,976\,x^7 + 200\,350\,x^8 + 655\,128\,x^9 + \\
& 2\,160\,946\,x^{10} + 7\,174\,284\,x^{11} + 23\,931\,431\,x^{12} + 80\,088\,232\,x^{13} + 268\,533\,346\,x^{14} + 900\,952\,616\,x^{15} + \\
& 3\,020\,792\,127\,x^{16} + 10\,108\,375\,872\,x^{17} + 33\,713\,509\,846\,x^{18} + 111\,923\,642\,152\,x^{19} + \\
& 369\,402\,521\,535\,x^{20} + 1\,210\,726\,388\,248\,x^{21} + 3\,936\,533\,037\,776\,x^{22} + 12\,685\,237\,008\,908\,x^{23} + \\
& 40\,478\,916\,453\,058\,x^{24} + 127\,808\,208\,168\,888\,x^{25} + 398\,987\,468\,942\,972\,x^{26} + \\
& 1\,230\,585\,274\,049\,316\,x^{27} + 3\,747\,158\,873\,944\,852\,x^{28} + 11\,256\,844\,788\,218\,744\,x^{29} + \\
& 33\,337\,904\,274\,628\,982\,x^{30} + 97\,262\,007\,724\,151\,920\,x^{31} + 279\,317\,402\,353\,493\,431\,x^{32} + \\
& 788\,961\,865\,442\,655\,472\,x^{33} + 2\,190\,050\,198\,656\,275\,982\,x^{34} + 5\,969\,097\,066\,666\,454\,856\,x^{35} + \\
& 15\,959\,243\,444\,320\,374\,789\,x^{36} + 41\,814\,842\,694\,335\,096\,992\,x^{37} + 107\,250\,360\,069\,946\,223\,626\,x^{38} + \\
& 268\,980\,688\,953\,973\,506\,220\,x^{39} + 658\,823\,005\,552\,616\,698\,670\,x^{40} + \\
& 1\,573\,911\,411\,364\,197\,904\,136\,x^{41} + 3\,662\,372\,148\,579\,433\,031\,528\,x^{42} + \\
& 8\,288\,810\,405\,618\,048\,303\,764\,x^{43} + 18\,218\,781\,312\,415\,364\,201\,910\,x^{44} + \\
& 38\,830\,283\,226\,215\,754\,590\,124\,x^{45} + 80\,122\,552\,318\,818\,478\,228\,542\,x^{46} + \\
& 159\,797\,441\,983\,911\,026\,766\,664\,x^{47} + 307\,546\,997\,966\,192\,908\,777\,092\,x^{48} + \\
& 570\,276\,129\,931\,118\,722\,910\,536\,x^{49} + 1\,017\,234\,556\,425\,044\,352\,980\,410\,x^{50} + \\
& 1\,742\,984\,912\,780\,508\,701\,928\,440\,x^{51} + 2\,865\,140\,929\,041\,232\,943\,907\,155\,x^{52} + \\
& 4\,513\,574\,916\,325\,198\,026\,399\,956\,x^{53} + 6\,809\,195\,410\,836\,894\,243\,048\,078\,x^{54} + \\
& 9\,833\,931\,639\,364\,180\,835\,118\,744\,x^{55} + 13\,598\,578\,719\,797\,129\,078\,347\,614\,x^{56} + \\
& 18\,019\,437\,235\,366\,047\,215\,968\,952\,x^{57} + 22\,915\,124\,558\,760\,125\,705\,720\,272\,x^{58} + \\
& 28\,029\,535\,640\,317\,584\,983\,898\,028\,x^{59} + 33\,076\,929\,313\,522\,902\,689\,493\,717\,x^{60} + \\
& 37\,795\,325\,569\,869\,805\,146\,555\,036\,x^{61} + 41\,990\,547\,883\,710\,550\,630\,755\,232\,x^{62} + \\
& 45\,557\,847\,775\,698\,543\,490\,868\,312\,x^{63} + 48\,478\,293\,994\,743\,202\,645\,045\,393\,x^{64} + \\
& 50\,796\,901\,046\,251\,716\,425\,044\,576\,x^{65} + 52\,593\,932\,270\,429\,371\,365\,546\,572\,x^{66} + \\
& 53\,959\,255\,268\,440\,876\,673\,654\,988\,x^{67} + 54\,974\,864\,530\,320\,519\,568\,625\,907\,x^{68} + \\
& 55\,706\,121\,745\,133\,313\,931\,062\,344\,x^{69} + 56\,199\,662\,997\,223\,937\,138\,206\,538\,x^{70} + \\
& 56\,485\,199\,011\,118\,156\,426\,142\,320\,x^{71} + 56\,578\,717\,186\,086\,829\,451\,888\,706\,x^{72} + \\
& 56\,485\,199\,011\,118\,156\,426\,142\,320\,x^{73} + 56\,199\,662\,997\,223\,937\,138\,206\,538\,x^{74} + \\
& 55\,706\,121\,745\,133\,313\,931\,062\,344\,x^{75} + 54\,974\,864\,530\,320\,519\,568\,625\,907\,x^{76} +
\end{aligned}
$$

$$53\,959\,255\,268\,440\,876\,673\,654\,988\,x^{77} + 52\,593\,932\,270\,429\,371\,365\,546\,572\,x^{78} +$$
$$50\,796\,901\,046\,251\,716\,425\,044\,576\,x^{79} + 48\,478\,293\,994\,743\,202\,645\,045\,393\,x^{80} +$$
$$45\,557\,847\,775\,698\,543\,490\,868\,312\,x^{81} + 41\,990\,547\,883\,710\,550\,630\,755\,232\,x^{82} +$$
$$37\,795\,325\,569\,869\,805\,146\,555\,036\,x^{83} + 33\,076\,929\,313\,522\,902\,689\,493\,717\,x^{84} +$$
$$28\,029\,535\,640\,317\,584\,983\,898\,028\,x^{85} + 22\,915\,124\,558\,760\,125\,705\,720\,272\,x^{86} +$$
$$18\,019\,437\,235\,366\,047\,215\,968\,952\,x^{87} + 13\,598\,578\,719\,797\,129\,078\,347\,614\,x^{88} +$$
$$9\,833\,931\,639\,364\,180\,835\,118\,744\,x^{89} + 6\,809\,195\,410\,836\,894\,243\,048\,078\,x^{90} +$$
$$4\,513\,574\,916\,325\,198\,026\,399\,956\,x^{91} + 2\,865\,140\,929\,041\,232\,943\,907\,155\,x^{92} +$$
$$1\,742\,984\,912\,780\,508\,701\,928\,440\,x^{93} + 1\,017\,234\,556\,425\,044\,352\,980\,410\,x^{94} +$$
$$570\,276\,129\,931\,118\,722\,910\,536\,x^{95} + 307\,546\,997\,966\,192\,908\,777\,092\,x^{96} +$$
$$159\,797\,441\,983\,911\,026\,766\,664\,x^{97} + 80\,122\,552\,318\,818\,478\,228\,542\,x^{98} +$$
$$38\,830\,283\,226\,215\,754\,590\,124\,x^{99} + 18\,218\,781\,312\,415\,364\,201\,910\,x^{100} +$$
$$8\,288\,810\,405\,618\,048\,303\,764\,x^{101} + 3\,662\,372\,148\,579\,433\,031\,528\,x^{102} +$$
$$1\,573\,911\,411\,364\,197\,904\,136\,x^{103} + 658\,823\,005\,552\,616\,698\,670\,x^{104} +$$
$$268\,980\,688\,953\,973\,506\,220\,x^{105} + 107\,250\,360\,069\,946\,223\,626\,x^{106} +$$
$$41\,814\,842\,694\,335\,096\,992\,x^{107} + 15\,959\,243\,444\,320\,374\,789\,x^{108} + 5\,969\,097\,066\,666\,454\,856\,x^{109} +$$
$$2\,190\,050\,198\,656\,275\,982\,x^{110} + 788\,961\,865\,442\,655\,472\,x^{111} + 279\,317\,402\,353\,493\,431\,x^{112} +$$
$$97\,262\,007\,724\,151\,920\,x^{113} + 33\,337\,904\,274\,628\,982\,x^{114} + 11\,256\,844\,788\,218\,744\,x^{115} +$$
$$3\,747\,158\,873\,944\,852\,x^{116} + 1\,230\,585\,274\,049\,316\,x^{117} + 398\,987\,468\,942\,972\,x^{118} +$$
$$127\,808\,208\,168\,888\,x^{119} + 40\,478\,916\,453\,058\,x^{120} + 12\,685\,237\,008\,908\,x^{121} +$$
$$3\,936\,533\,037\,776\,x^{122} + 1\,210\,726\,388\,248\,x^{123} + 369\,402\,521\,535\,x^{124} + 111\,923\,642\,152\,x^{125} +$$
$$33\,713\,509\,846\,x^{126} + 10\,108\,375\,872\,x^{127} + 3\,020\,792\,127\,x^{128} + 900\,952\,616\,x^{129} + 268\,533\,346\,x^{130} +$$
$$80\,088\,232\,x^{131} + 23\,931\,431\,x^{132} + 7\,174\,284\,x^{133} + 2\,160\,946\,x^{134} + 655\,128\,x^{135} + 200\,350\,x^{136} +$$
$$61\,976\,x^{137} + 19\,460\,x^{138} + 6244\,x^{139} + 2073\,x^{140} + 720\,x^{141} + 264\,x^{142} + 144\,x^{143} + x^{144}$$

$n = 7$

All computations have been done on radon1 (compute-grantley, Intel Xeon E5-2630v3 processors at 2.4 GHz).

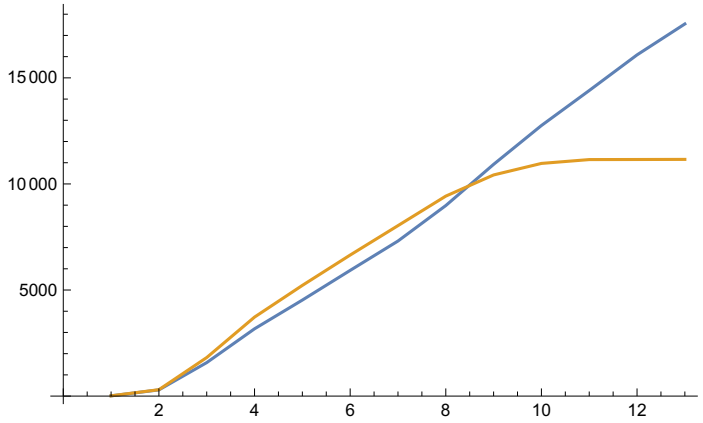
Old version (without discarding uninteresting states):
MatrixLabels[7] took 36s, matrix dimension is 1,848,612.
BuildMatrix[7] took about 46h using 15 parallel processes.
Matrix (SparseArray) has ByteCount 34,944,460,920 and 29G file size.
Matrix-vector multiplications take about 17h for a single prime (one processor).

New version (after discarding uninteresting states):
MatrixLabels[7] took 229s, matrix dimension is 393,878.
BuildMatrix[7] took about 8h using 15 parallel processes.
Matrix (SparseArray) has ByteCount 5,400,473,096 and 4.2G file size.
The matrix contains 337,332,580 nonzero entries, which is 0.2% of all entries.

Timings for the matrix-vector multiplications:
- computing over Z[x] without any tricks took 28.7h
- over Z[x] using PolynomialMod[..., x^99]: took 25h
- using evaluation-interpolation and Chinese remaindering: we used 5 primes , each took 2.5h (without exploiting palindromicity), but actually 4 primes would suffice.

In[65]:=
```
(* Comparing the timings for matrix-
 vector mults without (blue) and with (orange) PolynomialMod. *)
time1 = {12, 293, 1579, 3172, 4526, 5925, 7309,
   8981, 10 923, 12 753, 14 403, 16 083, 17 541};
time2 = {13, 302, 1817, 3724, 5218, 6648, 8035, 9426,
   10 424, 10 967, 11 148, 11 152, 11 157};
ListPlot[{time1, time2}, Joined → True]
```

Out[67]=



In[•]:=
```
res7 = Get["res7.m"]
```

Out[•]=

$$1 + 196\,x + 364\,x^2 + 1008\,x^3 + 2949\,x^4 + 9032\,x^5 + 28\,636\,x^6 + 92\,824\,x^7 + 305\,602\,x^8 +$$
$$1\,018\,488\,x^9 + 3\,427\,258\,x^{10} + 11\,621\,228\,x^{11} + 39\,646\,559\,x^{12} + 135\,916\,456\,x^{13} +$$
$$467\,742\,754\,x^{14} + 1\,614\,477\,056\,x^{15} + 5\,584\,888\,343\,x^{16} + 19\,348\,784\,116\,x^{17} +$$
$$67\,091\,184\,166\,x^{18} + 232\,687\,142\,592\,x^{19} + 806\,671\,850\,015\,x^{20} + 2\,793\,569\,995\,932\,x^{21} +$$
$$9\,657\,874\,655\,760\,x^{22} + 33\,311\,167\,589\,996\,x^{23} + 114\,557\,249\,974\,152\,x^{24} +$$
$$392\,584\,736\,221\,612\,x^{25} + 1\,339\,966\,269\,720\,544\,x^{26} + 4\,552\,957\,723\,269\,268\,x^{27} +$$
$$15\,393\,564\,937\,802\,680\,x^{28} + 51\,766\,856\,356\,719\,380\,x^{29} + 173\,086\,991\,762\,529\,204\,x^{30} +$$
$$575\,200\,600\,368\,228\,136\,x^{31} + 1\,899\,176\,367\,042\,221\,859\,x^{32} + 6\,228\,098\,764\,298\,730\,672\,x^{33} +$$
$$20\,278\,954\,430\,492\,886\,402\,x^{34} + 65\,537\,839\,196\,427\,520\,532\,x^{35} + 210\,160\,714\,227\,344\,572\,763\,x^{36} +$$
$$668\,465\,417\,072\,388\,948\,020\,x^{37} + 2\,108\,275\,713\,775\,443\,402\,792\,x^{38} +$$
$$6\,590\,936\,830\,062\,630\,268\,036\,x^{39} + 20\,416\,665\,319\,919\,204\,593\,976\,x^{40} +$$
$$62\,644\,455\,286\,175\,249\,799\,264\,x^{41} + 190\,317\,027\,427\,351\,248\,209\,426\,x^{42} +$$
$$572\,269\,642\,752\,615\,404\,988\,324\,x^{43} + 1\,702\,454\,912\,475\,548\,880\,416\,708\,x^{44} +$$
$$5\,008\,620\,772\,327\,221\,635\,476\,964\,x^{45} + 14\,565\,826\,972\,771\,182\,906\,835\,562\,x^{46} +$$
$$41\,852\,719\,486\,683\,325\,520\,878\,516\,x^{47} + 118\,760\,035\,536\,648\,112\,186\,596\,399\,x^{48} +$$
$$332\,621\,618\,717\,541\,007\,242\,625\,624\,x^{49} + 919\,026\,243\,745\,961\,707\,100\,737\,618\,x^{50} +$$
$$2\,503\,539\,111\,741\,912\,598\,371\,692\,760\,x^{51} + 6\,720\,000\,743\,010\,223\,225\,629\,459\,676\,x^{52} +$$
$$17\,762\,321\,485\,444\,188\,784\,607\,719\,688\,x^{53} + 46\,201\,923\,783\,840\,537\,548\,945\,803\,266\,x^{54} +$$

$$
\begin{aligned}
&118\,182\,250\,666\,462\,785\,295\,368\,572\,832\,x^{55} + 297\,074\,784\,098\,175\,638\,940\,354\,572\,453\,x^{56} + \\
&733\,293\,426\,951\,633\,848\,260\,072\,754\,676\,x^{57} + 1\,776\,042\,013\,933\,766\,860\,114\,555\,259\,894\,x^{58} + \\
&4\,217\,403\,821\,433\,856\,766\,164\,336\,942\,120\,x^{59} + 9\,810\,590\,662\,057\,480\,976\,162\,768\,103\,509\,x^{60} + \\
&22\,337\,447\,978\,303\,032\,318\,702\,082\,006\,316\,x^{61} + 49\,737\,085\,644\,351\,514\,079\,401\,039\,890\,090\,x^{62} + \\
&108\,204\,627\,218\,852\,567\,533\,147\,400\,267\,540\,x^{63} + 229\,790\,414\,758\,350\,040\,622\,881\,530\,491\,664\,x^{64} + \\
&475\,919\,433\,620\,005\,521\,592\,177\,187\,232\,072\,x^{65} + 960\,373\,750\,515\,797\,735\,901\,596\,038\,823\,190\,x^{66} + \\
&1\,886\,426\,144\,539\,755\,197\,542\,366\,950\,164\,676\,x^{67} + \\
&3\,603\,463\,562\,271\,470\,278\,766\,144\,720\,636\,919\,x^{68} + \\
&6\,687\,660\,377\,560\,681\,839\,127\,413\,260\,352\,360\,x^{69} + \\
&12\,047\,720\,553\,625\,765\,230\,285\,696\,585\,519\,978\,x^{70} + \\
&21\,048\,986\,191\,554\,081\,338\,271\,320\,501\,324\,396\,x^{71} + \\
&35\,636\,570\,870\,448\,160\,391\,825\,939\,394\,869\,241\,x^{72} + \\
&58\,421\,779\,892\,277\,898\,321\,966\,025\,872\,861\,636\,x^{73} + \\
&92\,680\,265\,573\,275\,883\,066\,462\,543\,725\,492\,320\,x^{74} + \\
&142\,203\,115\,526\,883\,375\,986\,681\,843\,994\,589\,196\,x^{75} + \\
&210\,953\,455\,979\,273\,177\,212\,443\,099\,311\,309\,719\,x^{76} + \\
&302\,518\,982\,189\,001\,643\,600\,161\,148\,257\,937\,612\,x^{77} + \\
&419\,414\,266\,663\,605\,070\,074\,068\,835\,210\,833\,144\,x^{78} + \\
&562\,360\,885\,598\,155\,685\,855\,263\,942\,304\,236\,072\,x^{79} + \\
&729\,729\,781\,326\,919\,550\,706\,954\,110\,024\,766\,073\,x^{80} + \\
&917\,336\,030\,209\,541\,784\,252\,930\,471\,339\,022\,420\,x^{81} + \\
&1\,118\,711\,270\,198\,145\,074\,669\,404\,699\,590\,215\,528\,x^{82} + \\
&1\,325\,852\,432\,638\,170\,369\,255\,136\,520\,479\,847\,528\,x^{83} + \\
&1\,530\,298\,891\,298\,052\,558\,927\,424\,293\,178\,308\,270\,x^{84} + \\
&1\,724\,282\,810\,724\,736\,634\,690\,527\,630\,305\,918\,288\,x^{85} + \\
&1\,901\,676\,592\,646\,601\,662\,521\,560\,701\,456\,849\,748\,x^{86} + \\
&2\,058\,536\,107\,050\,470\,915\,075\,406\,947\,408\,057\,076\,x^{87} + \\
&2\,193\,174\,336\,388\,506\,016\,870\,609\,750\,081\,706\,125\,x^{88} + \\
&2\,305\,838\,124\,320\,238\,406\,934\,170\,994\,712\,839\,112\,x^{89} + \\
&2\,398\,148\,990\,859\,061\,766\,752\,433\,948\,220\,010\,940\,x^{90} + \\
&2\,472\,486\,467\,185\,862\,271\,462\,062\,816\,830\,965\,980\,x^{91} + \\
&2\,531\,452\,168\,053\,649\,184\,750\,335\,185\,701\,984\,144\,x^{92} + \\
&2\,577\,486\,829\,393\,026\,562\,419\,600\,955\,011\,378\,644\,x^{93} + \\
&2\,612\,651\,489\,221\,830\,492\,656\,346\,599\,025\,536\,818\,x^{94} + \\
&2\,638\,544\,582\,848\,345\,834\,713\,330\,508\,570\,024\,132\,x^{95} + \\
&2\,656\,310\,541\,191\,193\,017\,428\,479\,173\,140\,918\,061\,x^{96} + \\
&2\,666\,695\,451\,419\,442\,057\,962\,747\,410\,658\,373\,480\,x^{97} + \\
&2\,670\,113\,158\,846\,160\,742\,372\,913\,777\,087\,464\,324\,x^{98} + \\
&2\,666\,695\,451\,419\,442\,057\,962\,747\,410\,658\,373\,480\,x^{99} + \\
&2\,656\,310\,541\,191\,193\,017\,428\,479\,173\,140\,918\,061\,x^{100} + \\
&2\,638\,544\,582\,848\,345\,834\,713\,330\,508\,570\,024\,132\,x^{101} + \\
&2\,612\,651\,489\,221\,830\,492\,656\,346\,599\,025\,536\,818\,x^{102} + \\
&2\,577\,486\,829\,393\,026\,562\,419\,600\,955\,011\,378\,644\,x^{103} + \\
&2\,531\,452\,168\,053\,649\,184\,750\,335\,185\,701\,984\,144\,x^{104} +
\end{aligned}
$$

$$2\,472\,486\,467\,185\,862\,271\,462\,062\,816\,830\,965\,980\,x^{105} +$$
$$2\,398\,148\,990\,859\,061\,766\,752\,433\,948\,220\,010\,940\,x^{106} +$$
$$2\,305\,838\,124\,320\,238\,406\,934\,170\,994\,712\,839\,112\,x^{107} +$$
$$2\,193\,174\,336\,388\,506\,016\,870\,609\,750\,081\,706\,125\,x^{108} +$$
$$2\,058\,536\,107\,050\,470\,915\,075\,406\,947\,408\,057\,076\,x^{109} +$$
$$1\,901\,676\,592\,646\,601\,662\,521\,560\,701\,456\,849\,748\,x^{110} +$$
$$1\,724\,282\,810\,724\,736\,634\,690\,527\,630\,305\,918\,288\,x^{111} +$$
$$1\,530\,298\,891\,298\,052\,558\,927\,424\,293\,178\,308\,270\,x^{112} +$$
$$1\,325\,852\,432\,638\,170\,369\,255\,136\,520\,479\,847\,528\,x^{113} +$$
$$1\,118\,711\,270\,198\,145\,074\,669\,404\,699\,590\,215\,528\,x^{114} +$$
$$917\,336\,030\,209\,541\,784\,252\,930\,471\,339\,022\,420\,x^{115} +$$
$$729\,729\,781\,326\,919\,550\,706\,954\,110\,024\,766\,073\,x^{116} +$$
$$562\,360\,885\,598\,155\,685\,855\,263\,942\,304\,236\,072\,x^{117} +$$
$$419\,414\,266\,663\,605\,070\,074\,068\,835\,210\,833\,144\,x^{118} +$$
$$302\,518\,982\,189\,001\,643\,600\,161\,148\,257\,937\,612\,x^{119} +$$
$$210\,953\,455\,979\,273\,177\,212\,443\,099\,311\,309\,719\,x^{120} +$$
$$142\,203\,115\,526\,883\,375\,986\,681\,843\,994\,589\,196\,x^{121} +$$
$$92\,680\,265\,573\,275\,883\,066\,462\,543\,725\,492\,320\,x^{122} +$$
$$58\,421\,779\,892\,277\,898\,321\,966\,025\,872\,861\,636\,x^{123} +$$
$$35\,636\,570\,870\,448\,160\,391\,825\,939\,394\,869\,241\,x^{124} +$$
$$21\,048\,986\,191\,554\,081\,338\,271\,320\,501\,324\,396\,x^{125} +$$
$$12\,047\,720\,553\,625\,765\,230\,285\,696\,585\,519\,978\,x^{126} +$$
$$6\,687\,660\,377\,560\,681\,839\,127\,413\,260\,352\,360\,x^{127} +$$
$$3\,603\,463\,562\,271\,470\,278\,766\,144\,720\,636\,919\,x^{128} +$$
$$1\,886\,426\,144\,539\,755\,197\,542\,366\,950\,164\,676\,x^{129} +$$
$$960\,373\,750\,515\,797\,735\,901\,596\,038\,823\,190\,x^{130} +$$
$$475\,919\,433\,620\,005\,521\,592\,177\,187\,232\,072\,x^{131} +$$
$$229\,790\,414\,758\,350\,040\,622\,881\,530\,491\,664\,x^{132} +$$
$$108\,204\,627\,218\,852\,567\,533\,147\,400\,267\,540\,x^{133} +$$
$$49\,737\,085\,644\,351\,514\,079\,401\,039\,890\,090\,x^{134} + 22\,337\,447\,978\,303\,032\,318\,702\,082\,006\,316\,x^{135} +$$
$$9\,810\,590\,662\,057\,480\,976\,162\,768\,103\,509\,x^{136} + 4\,217\,403\,821\,433\,856\,766\,164\,336\,942\,120\,x^{137} +$$
$$1\,776\,042\,013\,933\,766\,860\,114\,555\,259\,894\,x^{138} + 733\,293\,426\,951\,633\,848\,260\,072\,754\,676\,x^{139} +$$
$$297\,074\,784\,098\,175\,638\,940\,354\,572\,453\,x^{140} + 118\,182\,250\,666\,462\,785\,295\,368\,572\,832\,x^{141} +$$
$$46\,201\,923\,783\,840\,537\,548\,945\,803\,266\,x^{142} + 17\,762\,321\,485\,444\,188\,784\,607\,719\,688\,x^{143} +$$
$$6\,720\,000\,743\,010\,223\,225\,629\,459\,676\,x^{144} + 2\,503\,539\,111\,741\,912\,598\,371\,692\,760\,x^{145} +$$
$$919\,026\,243\,745\,961\,707\,100\,737\,618\,x^{146} + 332\,621\,618\,717\,541\,007\,242\,625\,624\,x^{147} +$$
$$118\,760\,035\,536\,648\,112\,186\,596\,399\,x^{148} + 41\,852\,719\,486\,683\,325\,520\,878\,516\,x^{149} +$$
$$14\,565\,826\,972\,771\,182\,906\,835\,562\,x^{150} + 5\,008\,620\,772\,327\,221\,635\,476\,964\,x^{151} +$$
$$1\,702\,454\,912\,475\,548\,880\,416\,708\,x^{152} + 572\,269\,642\,752\,615\,404\,988\,324\,x^{153} +$$
$$190\,317\,027\,427\,351\,248\,209\,426\,x^{154} + 62\,644\,455\,286\,175\,249\,799\,264\,x^{155} +$$
$$20\,416\,665\,319\,919\,204\,593\,976\,x^{156} + 6\,590\,936\,830\,062\,630\,268\,036\,x^{157} +$$
$$2\,108\,275\,713\,775\,443\,402\,792\,x^{158} + 668\,465\,417\,072\,388\,948\,020\,x^{159} +$$
$$210\,160\,714\,227\,344\,572\,763\,x^{160} + 65\,537\,839\,196\,427\,520\,532\,x^{161} +$$
$$20\,278\,954\,430\,492\,886\,402\,x^{162} + 6\,228\,098\,764\,298\,730\,672\,x^{163} +$$

$$1\,899\,176\,367\,042\,221\,859\,x^{164} + 575\,200\,600\,368\,228\,136\,x^{165} + 173\,086\,991\,762\,529\,204\,x^{166} + 51\,766\,856\,356\,719\,380\,x^{167} + 15\,393\,564\,937\,802\,680\,x^{168} + 4\,552\,957\,723\,269\,268\,x^{169} + 1\,339\,966\,269\,720\,544\,x^{170} + 392\,584\,736\,221\,612\,x^{171} + 114\,557\,249\,974\,152\,x^{172} + 33\,311\,167\,589\,996\,x^{173} + 9\,657\,874\,655\,760\,x^{174} + 2\,793\,569\,995\,932\,x^{175} + 806\,671\,850\,015\,x^{176} + 232\,687\,142\,592\,x^{177} + 67\,091\,184\,166\,x^{178} + 19\,348\,784\,116\,x^{179} + 5\,584\,888\,343\,x^{180} + 1\,614\,477\,056\,x^{181} + 467\,742\,754\,x^{182} + 135\,916\,456\,x^{183} + 39\,646\,559\,x^{184} + 11\,621\,228\,x^{185} + 3\,427\,258\,x^{186} + 1\,018\,488\,x^{187} + 305\,602\,x^{188} + 92\,824\,x^{189} + 28\,636\,x^{190} + 9032\,x^{191} + 2949\,x^{192} + 1008\,x^{193} + 364\,x^{194} + 196\,x^{195} + x^{196}$$

Sequence terms

In[•]:=
```
{4, 140, 161 036, 14 314 228 378, 99 704 315 229 167 288,
  56 578 717 186 086 829 451 888 706, 2 670 113 158 846 160 742 372 913 777 087 464 324} / 2
```

Out[•]=
```
{2, 70, 80 518, 7 157 114 189, 49 852 157 614 583 644,
 28 289 358 593 043 414 725 944 353, 1 335 056 579 423 080 371 186 456 888 543 732 162}
```