In[ ]:=
```
(*Utilities*)
Once[<< KnotTheory`]
RotateToMinimal[l_] := Module[
   {bestl = l, rotatedl = RotateLeft[l]},
   While[rotatedl =!= l,
    bestl = First[Sort[{bestl, rotatedl}]];
    rotatedl = RotateLeft[rotatedl]
   ];
   bestl
  ];

SignX[c_X] := If[PositiveQ[c], 1, -1] (*determines sign of crossing*)
FindJointCrossing[pd_, m_, n_] :=
 (*Finds the unique?! crossing containing both m and n*)
 Module[{Int},
  Int = Intersection[#[[1]] & /@ Position[pd, m], #[[1]] & /@ Position[pd, n]];
  If[Length[Int] == 1, Int[[1]], If[Length[Int] == 0, Pi,
    Print["WARNING:two or more crossings contain both m and n"]]]
 ]
FindJointCrossing2[pd_, m_, n_] :=
 (*Finds the unique?! crossing containing both m and n such that m,
 n are part of a cycle*)
 Module[{Int, incycle = False},
  Int = Intersection[#[[1]] & /@ Position[pd, m], #[[1]] & /@ Position[pd, n]];
  If[Length[Int] == 1, Int = Int[[1]];
   If[SignX[pd[[Int]]] == -1, If[MemberQ[{Sort[{pd[[Int]][[2]], pd[[Int]][[3]]}],
       Sort[{pd[[Int]][[1]], pd[[Int]][[4]]}]}, Sort[{m, n}]], incycle = True]];
   If[SignX[pd[[Int]]] == 1,
    {Sort[{pd[[Int]][[1]], pd[[Int]][[2]]}], Sort[{pd[[Int]][[3]], pd[[Int]][[4]]}]};
    If[MemberQ[{Sort[{pd[[Int]][[1]], pd[[Int]][[2]]}], Sort[{pd[[Int]][[3]], pd[[Int]][[4]]}]},
      Sort[{m, n}]], incycle = True]];
  ];
  incycle

 ]
```

```
Loading KnotTheory` version of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
```

A function that takes input a PD code PD[X[a,b,c,d],X[e,f,g,h],...] and has output a tangle in the form {Matching,Garsides} where Garsides encodes a braid on strands 1,.., 2b by listing the Garside elements and their sign
in order (top to bottom). Each Garside is a special braid word but also includes half-twists on each of the participating strands.
For example Garside could contain an element $G_1[3, 4, 5]$ which means $\sigma_3\,\sigma_4\,\sigma_5\,\sigma_3\,\sigma_4\sigma_3$ composed with half twists on strands 3,4,5
and $G_{-1}[3, 4, 5]$ would be its inverse.

Matching matches the 2b strands on top and should be projected as caps that intersect but in

a way such that the first matching is on above then the next and so on. This order is arbitrary and does not affect the tangle.

Matching also contains a sign for each cap which is interpreted as a half-twist on that strand.

In[•]:=

```
Knot2Bandtangle[pd_] :=
 Module[{cycles, bl, safety = 0, treecrossings = {}, othercrossings, Garsides = {},
   caps = {}, bands, matchings, m},
  cycles = (Times @@ (pd /. X[i_, j_, k_, l_] :>
           If[PositiveQ@X[i, j, k, l], cyc[i, j] cyc[l, k], cyc[i, l] cyc[j, k]])) //.
     cyc[i_, m1___, j_] cyc[j_, m2___, k_] :> cyc[i, m1, j, m2, k] /.
    c_cyc :> RotateToMinimal@Most[c];
  cycles = SortBy[List @@ cycles, First];

  bl = cycles;
  Garsides = {};
  caps = {};
  While[Length[bl] > 1 && safety < 100,
   safety++;
   m = bl[[-1, 1]];
   treecrossings = Append[treecrossings, FindJointCrossing[pd, m, m - 1]];
   sign = SignX@pd[[FindJointCrossing[pd, m, m - 1]]];
   Garsides = Append[Garsides, G_sign[Sequence @@ Reverse@Last@bl]];
   (*G for Garside*)
   (*bl=Delete[bl,FirstPosition[bl,m-1]+{0,1}];*)
   bl = Most[bl] /. m - 1 → Sequence @@ Prepend[Reverse@Last@bl, m - 1];
  ];
  bl = Append[List @@ (bl[[1]]), 1];
  othercrossings = Complement[Range@Length[pd], treecrossings];

  bands = Table[If[FindJointCrossing2[Part[pd, othercrossings], bl[[i]], bl[[i + 1]]],
     {bl[[i]], bl[[i + 1]]}, Nothing], {i, 1, Length[bl] - 1}];

  matchings = {FindJointCrossing[Part[pd, othercrossings], #[[1]], #[[2]],
      -SignX@Part[Part[pd, othercrossings],
         FindJointCrossing[Part[pd, othercrossings], #[[1]], #[[2]]]]} & /@ bands;
  matchings = matchings /. {a_, b_} → Subscript[a, b];

  Garsides = Sort /@ (Garsides /. Thread[bl → Range@Length@bl]);
  bf[FF_] := Table[{FF[[i]], FF[[i + 1]]}, {i, 1, Length[FF] - 1}];
  Garsides = (G_#[[0,2]] @@ (bf@(List @@ #))) & /@ Garsides;
  bands = bands /. Thread[bl → Range@Length@bl];
  Garsides = (G_#[[0,2]] @@ (Intersection[(List @@ #), bands])) & /@ Garsides;
  Thread[bands → Range@Length@bands];
  Garsides = Garsides /. Thread[bands → Range@Length@bands];
  {Garsides, matchings}
 ]
```

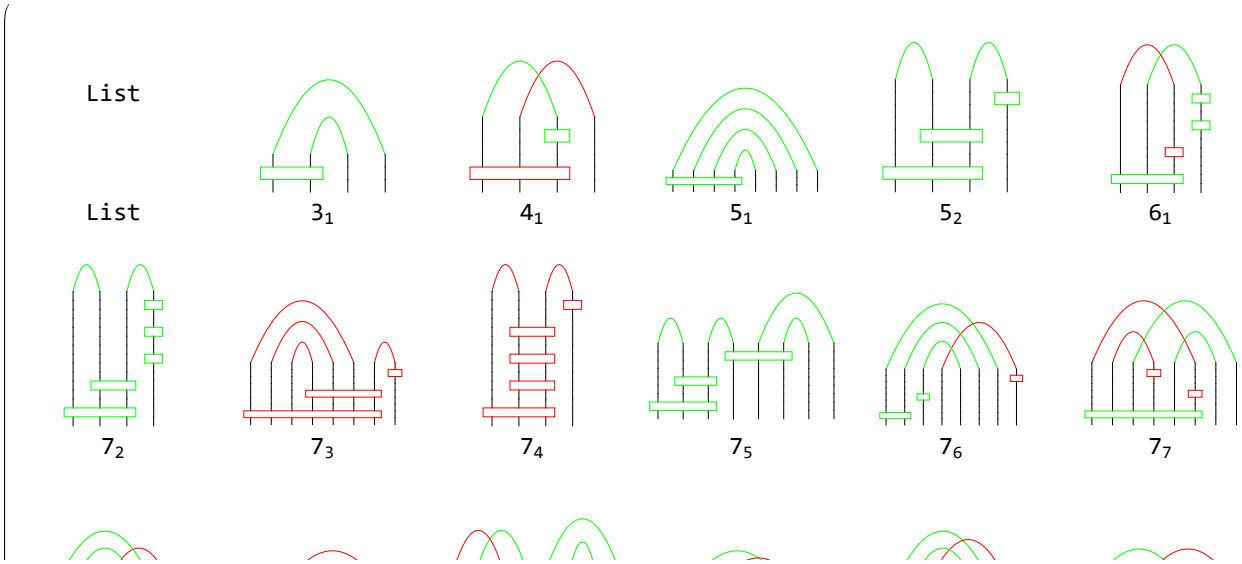
In[•]:=
```
Knot2Bandtangle[PD@Knot[5, 1]]
```

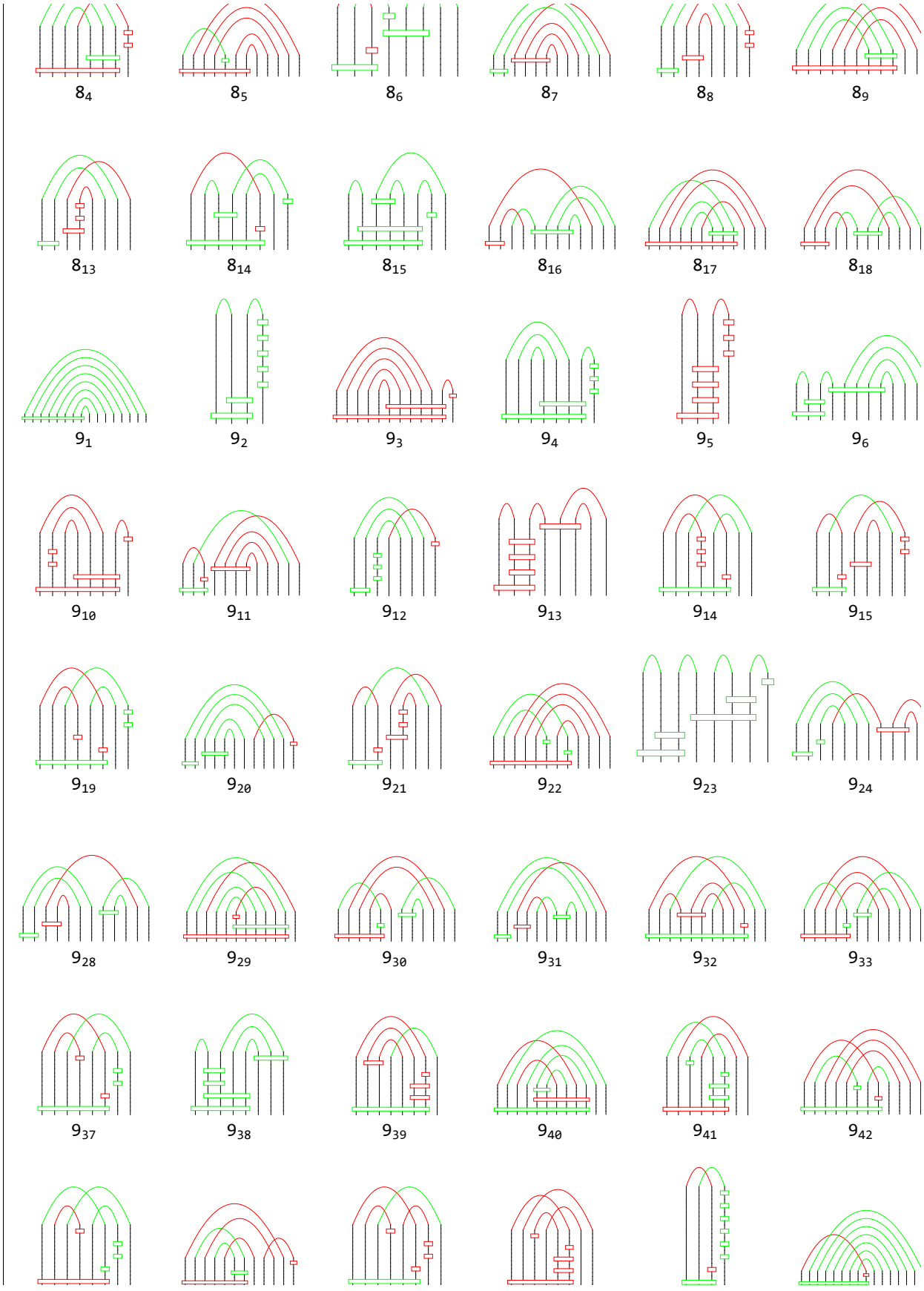
Out[•]=
```
{{G1[1, 2, 3, 4]}, {21, 41, 11, 31, 31, 11, 41, 21}}
```

In[•]:=
```
(*A routine for visualizing our output. Green is positive,
red negative. The boxes are Garsides (with half twists on each strand!)
 The input is a list of Garsides with signs and a matching with signs. The color of
 the matching arcs denotes the sign of the half twist on that matching arc.*)
DrawTangle[{GG_, M_}] :=
 Graphics[{If[Length[#[[2]]] == 2, {If[#[[1]] == 1, Green, Red], BezierCurve[{{#[[2, 1, 1]], 0},
          {(#[[2, 1, 1]] + #[[2, 2, 1]]) / 2, Abs[#[[2, 1, 1]] - #[[2, 2, 1]]] + 1}, {#[[2, 2, 1]], 0}}]}
       , Nothing] & /@ Flatten[Table[{{1, Position[M, i1]}, {-1, Position[M, i-1]}},
       {i, Length[M] / 2}], 1],
   Table[
    {Table[Line[{{k, -i + 1}, {k, -i - 1 / 3 + 1}}], {k, Length[M]}],
     Table[Line[{{k, -i - 2 / 3 + 1}, {k, -i - 1 + 1}}], {k, Length[M]}],
     Table[Line[{{k, -i - 1 / 3 + 1}, {k, -i - 2 / 3 + 1}}],
      {k, Complement[Range@Length[M], List @@ GG[[i]]]}],
     {If[GG[[i, 0, 2]] == 1, Green, Red], Line[{{First@GG[[i]] - 1 / 3, -i - 1 / 3 + 1},
        {Last@GG[[i]] + 1 / 3, -i - 1 / 3 + 1}, {Last@GG[[i]] + 1 / 3, -i - 2 / 3 + 1},
        {First@GG[[i]] - 1 / 3, -i - 2 / 3 + 1}, {First@GG[[i]] - 1 / 3, -i - 1 / 3 + 1}}]}
    }
    , {i, Length[GG]}]
   }
   , ImageSize → Tiny]
(*and something to wrap a vector around for nicer display*)
WrapVectors[V_, H_, w_] := Table[If[EvenQ[i], If[w i / 2 + j ≤ Length[V], V[[w i / 2 + j]]],
   If[w (i - 1) / 2 + j ≤ Length[V], H[[w (i - 1) / 2 + j]]]], {i, 0, 2 Length[V] / w}, {j, 0, w}]
```

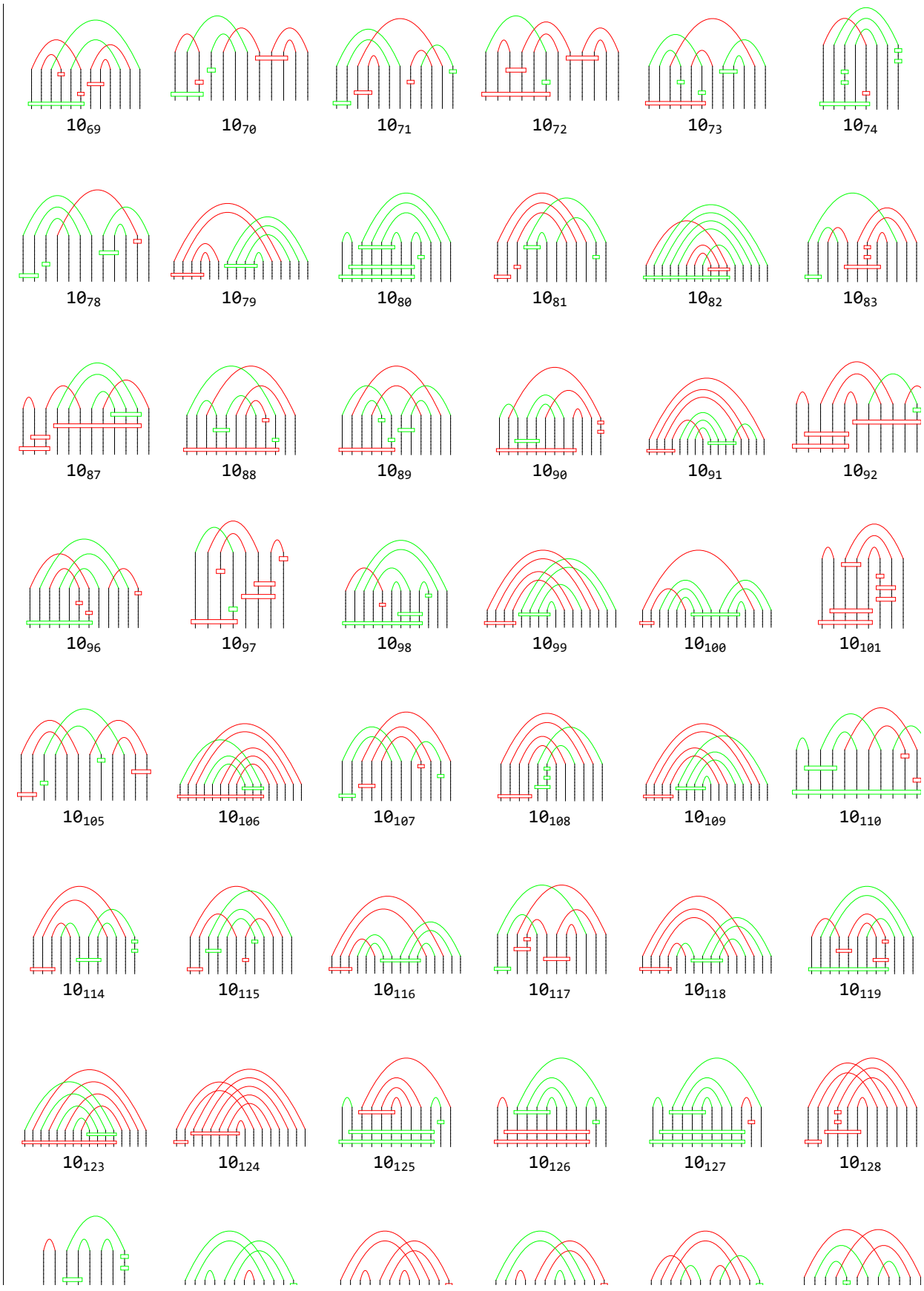
In[•]:=
```
WrapVectors[DrawTangle /@ Knot2Bandtangle /@ PD /@ AllKnots[{3, 10}],
  Subscript @@ # & /@ AllKnots[{3, 10}], 9] // MatrixForm
```

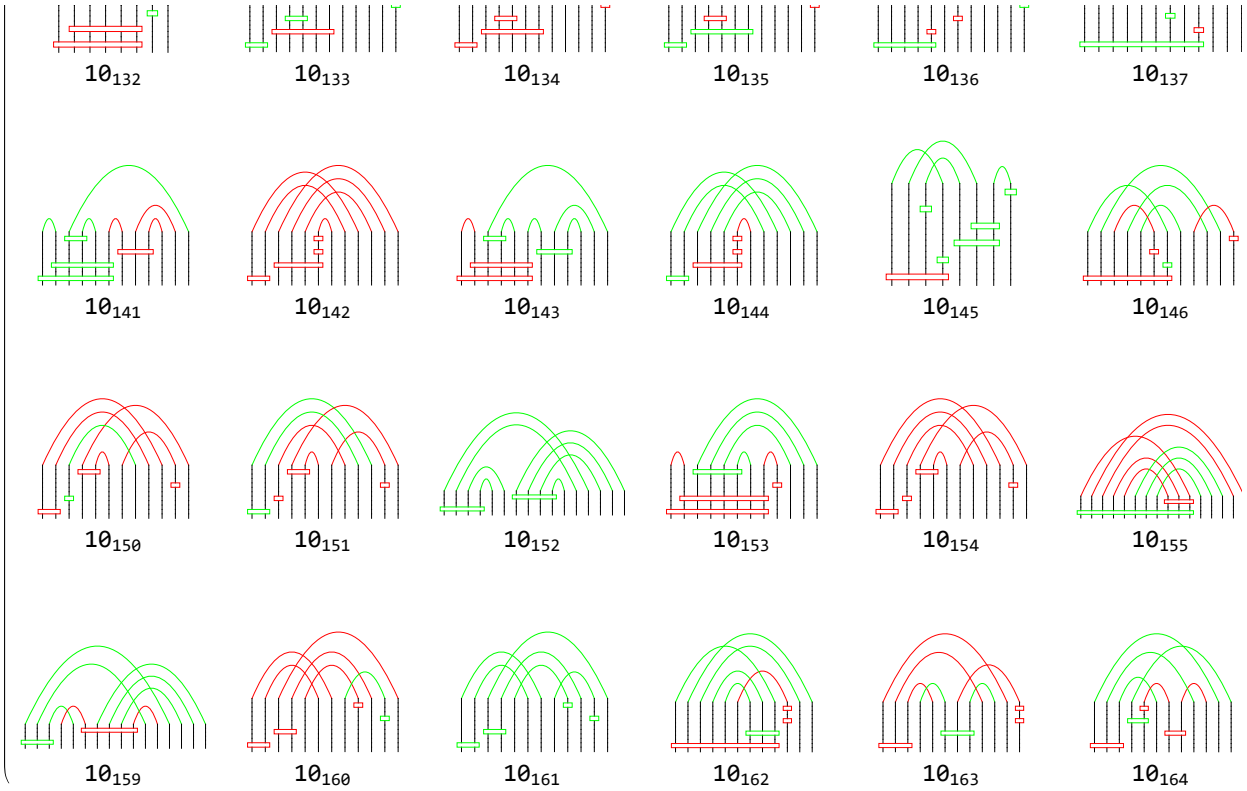
Out[•]//MatrixForm=




8_4
8_5
8_6
8_7
8_8
8_9
8_{13}
8_{14}
8_{15}
8_{16}
8_{17}
8_{18}
9_1
9_2
9_3
9_4
9_5
9_6
9_{10}
9_{11}
9_{12}
9_{13}
9_{14}
9_{15}
9_{19}
9_{20}
9_{21}
9_{22}
9_{23}
9_{24}
9_{28}
9_{29}
9_{30}
9_{31}
9_{32}
9_{33}
9_{37}
9_{38}
9_{39}
9_{40}
9_{41}
9_{42}


9_{46}
9_{47}
9_{48}
9_{49}
10_{1}
10_{2}
10_{6}
10_{7}
10_{8}
10_{9}
10_{10}
10_{11}
10_{15}
10_{16}
10_{17}
10_{18}
10_{19}
10_{20}
10_{24}
10_{25}
10_{26}
10_{27}
10_{28}
10_{29}
10_{33}
10_{34}
10_{35}
10_{36}
10_{37}
10_{38}
10_{42}
10_{43}
10_{44}
10_{45}
10_{46}
10_{47}
10_{51}
10_{52}
10_{53}
10_{54}
10_{55}
10_{56}
10_{60}
10_{61}
10_{62}
10_{63}
10_{64}
10_{65}


10_{69}
10_{70}
10_{71}
10_{72}
10_{73}
10_{74}
10_{78}
10_{79}
10_{80}
10_{81}
10_{82}
10_{83}
10_{87}
10_{88}
10_{89}
10_{90}
10_{91}
10_{92}
10_{96}
10_{97}
10_{98}
10_{99}
10_{100}
10_{101}
10_{105}
10_{106}
10_{107}
10_{108}
10_{109}
10_{110}
10_{114}
10_{115}
10_{116}
10_{117}
10_{118}
10_{119}
10_{123}
10_{124}
10_{125}
10_{126}
10_{127}
10_{128}



In[ ]:=
```
Knot2Bandtangle[PD@Knot[8, 20]]
```

Out[ ]=
```
{{G1[3, 4], G1[7], G-1[2, 3, 4, 5, 6], G-1[1, 2, 3, 4, 5, 6]}, {1-1, 1-1, 21, 31, 31, 41, 41, 21}}
```

Produce from a matching M a PD code of crossings that implement the matching in such a way that reading from left to right each new cap is strictly below the previously encountered caps.
Note all crossings are positive.
This is because the matching is supposed to come from a chord diagram with no crossings, just chords inside and outside the circle. Folding them upwards creates the crossings mentioned here.
The input is a list M of even length with every symbol appearing precisely twice.

Garside2Braid maps a Garside $G_s[a, b, c, d, ..]$ where a,b,c,d... is any sequence and s is the sign +1,-1 to a braid. The braid is oriented downwards

In[•]:=
```
Garside2Braid[Gar_] := Module[{sign = Gar[[0, 2]], places = List @@ Gar, ans},
  {places, sign};
  If[sign == -1, (*Also works if sign is positive but not 1*)
   ans = Table[
     X[c[places[[j + i]]] + j,
      c[places[[j]]] + j + i - 1, c[places[[j + i]]] + j + 1, c[places[[j]]] + j + i]
     , {j, 1, Length[places] - 1}, {i, 1, Length[places] - j}],
   ans = Table[
     X[c[places[[j]]] + j + i - 1,
      c[places[[j + i]]] + j + 1, c[places[[j]]] + j + i, c[places[[j + i]]] + j]
     , {j, 1, Length[places] - 1}, {i, 1, Length[places] - j}]
  ];
  Flatten[ans, 1] /. c[u_] → c[-u]
 ]
Garside2Braid[G_-1[3, 4]]
Garside2Braid[G_-1[3, 4, 5]]
```

Out[•]=
```
{X[1 + c[-4], 1 + c[-3], 2 + c[-4], 2 + c[-3]]}
```

Out[•]=
```
{X[1 + c[-4], 1 + c[-3], 2 + c[-4], 2 + c[-3]],
 X[1 + c[-5], 2 + c[-3], 2 + c[-5], 3 + c[-3]], X[2 + c[-5], 2 + c[-4], 3 + c[-5], 3 + c[-4]]}
```

Maps a list of Garsides to a big braid. Strands are named 1,2,..Nstrands from the top and every strand has a counter stored in counters counting how many crossings it crossed. Similarly we also keep track of the signed sum of crossings in halftwists. The braid is oriented downwards

In[•]:=
```
Garsides2Braid[Garsides_, Nstrands_] :=
 Module[{swapplaces, pd = {}, WG = Garsides, gars = 0, currentG,
   bottomline = Range[Nstrands], halftwists = ConstantArray[0, Nstrands], s},
  While[gars < 100 && Length[WG] > 0,
   gars++;
   currentG = First@WG;
   swapplaces = List @@ currentG;
   (*Print["Garside2Braid[",currentG];*)
   (*Garside2Braid[currentG];*)
   pd = Join[pd, Garside2Braid[currentG]];

   For[s = Min@swapplaces, s <= Max@swapplaces, s++,
    pd = m_{s,-s→s}[pd];
    halftwists[[s]] = halftwists[[s]] + Sign[currentG[[0, 2]]] (*Careful the 1 in G_1 is
     affected by the replacement rule three lines down from here, hence Sign.*)
   ];
   WG = (Rest@WG) /. Thread[swapplaces -> Reverse[swapplaces]];
   bottomline = bottomline /. Thread[swapplaces -> Reverse[swapplaces]];
  ];
  {pd, bottomline, halftwists}
 ]
Garsides2Braid[{G_1[3, 4], G_1[7], G_-1[2, 3, 4, 5, 6], G_-1[1, 2, 3, 4, 5, 6]}, 8]
Garsides2Braid[{G_-1[2, 3, 4, 5, 6], G_-1[1, 2, 3, 4, 5, 6]}, 8]
```

... Join: Heads $m_{4,-4\to 4}$ and List at positions 1 and 2 are expected to be the same.

... Join: Heads $m_{7,-7\to 7}$ and List at positions 1 and 2 are expected to be the same.

... Join: Heads $m_{6,-6\to 6}$ and List at positions 1 and 2 are expected to be the same.

... General: Further output of Join::heads will be suppressed during this calculation.

Out[•]=
```
{m6,-6→6[
  m5,-5→5[m4,-4→4[m3,-3→3[m2,-2→2[m1,-1→1[Join[m6,-6→6[m5,-5→5[m4,-4→4[m3,-3→3[m2,-2→2[Join[m7,-7→7[
                    Join[m4,-4→4[m3,-3→3[{X[1+c[-3],2+c[-4],2+c[-3],1+c[-4]]}]],{}]],
                  {X[1+c[-4],1+c[-2],2+c[-4],2+c[-2]],X[1+c[-3],
                    2+c[-2],2+c[-3],3+c[-2]],X[1+c[-5],3+c[-2],2+c[-5],
                    4+c[-2]],X[1+c[-6],4+c[-2],2+c[-6],5+c[-2]],X[2+c[-3],
                    2+c[-4],3+c[-3],3+c[-4]],X[2+c[-5],3+c[-4],3+c[-5],
                    4+c[-4]],X[2+c[-6],4+c[-4],3+c[-6],5+c[-4]],X[3+c[-5],
                    3+c[-3],4+c[-5],4+c[-3]],X[3+c[-6],4+c[-3],4+c[-6],
                    5+c[-3]],X[4+c[-6],4+c[-5],5+c[-6],5+c[-5]]}]]]]]]],
          {X[1+c[-6],1+c[-1],2+c[-6],2+c[-1]],X[1+c[-5],2+c[-1],
            2+c[-5],3+c[-1]],X[1+c[-3],3+c[-1],2+c[-3],4+c[-1]],
           X[1+c[-4],4+c[-1],2+c[-4],5+c[-1]],
           X[1+c[-2],5+c[-1],2+c[-2],6+c[-1]],
           X[2+c[-5],2+c[-6],3+c[-5],3+c[-6]],
           X[2+c[-3],3+c[-6],3+c[-3],4+c[-6]],
           X[2+c[-4],4+c[-6],3+c[-4],5+c[-6]],
           X[2+c[-2],5+c[-6],3+c[-2],6+c[-6]],
           X[3+c[-3],3+c[-5],4+c[-3],4+c[-5]],
           X[3+c[-4],4+c[-5],4+c[-4],5+c[-5]],
           X[3+c[-2],5+c[-5],4+c[-2],6+c[-5]],
           X[4+c[-4],4+c[-3],5+c[-4],5+c[-3]],
           X[4+c[-2],5+c[-3],5+c[-2],6+c[-3]],
           X[5+c[-2],5+c[-4],6+c[-2],6+c[-4]]}]]]]]]]],
 {2,4,3,5,6,1,7,8},{-1,-2,
  -1,
  -1,
  -2,
  -2,
  1,
  0}}
```

... Join: Heads $m_{6,-6\to 6}$ and List at positions 1 and 2 are expected to be the same.

Out[•]=
```
{m6,-6→6[
  m5,-5→5[m4,-4→4[m3,-3→3[m2,-2→2[m1,-1→1[Join[m6,-6→6[m5,-5→5[m4,-4→4[m3,-3→3[m2,-2→2[{X[1+c[-3],
                  1+c[-2],2+c[-3],2+c[-2]],X[1+c[-4],2+c[-2],2+c[-4],3+c[-2]],
                X[1+c[-5],3+c[-2],2+c[-5],4+c[-2]],X[1+c[-6],4+c[-2],2+c[-6],
                 5+c[-2]],X[2+c[-4],2+c[-3],3+c[-4],3+c[-3]],X[2+c[-5],3+
                  c[-3],3+c[-5],4+c[-3]],X[2+c[-6],4+c[-3],3+c[-6],5+c[-3]],
                X[3+c[-5],3+c[-4],4+c[-5],4+c[-4]],X[3+c[-6],4+c[-4],4+
                  c[-6],5+c[-4]],X[4+c[-6],4+c[-5],5+c[-6],5+c[-5]]}]]]]],
          {X[1+c[-6],1+c[-1],2+c[-6],2+c[-1]],X[1+c[-5],2+c[-1],
            2+c[-5],3+c[-1]],X[1+c[-4],3+c[-1],2+c[-4],4+c[-1]],
           X[1+c[-3],4+c[-1],2+c[-3],5+c[-1]],
           X[1+c[-2],5+c[-1],2+c[-2],6+c[-1]],
           X[2+c[-5],2+c[-6],3+c[-5],3+c[-6]],
           X[2+c[-4],3+c[-6],3+c[-4],4+c[-6]],
           X[2+c[-3],4+c[-6],3+c[-3],5+c[-6]],
           X[2+c[-2],5+c[-6],3+c[-2],6+c[-6]],
           X[3+c[-4],3+c[-5],4+c[-4],4+c[-5]],
           X[3+c[-3],4+c[-5],4+c[-3],5+c[-5]],
           X[3+c[-2],5+c[-5],4+c[-2],6+c[-5]],
           X[4+c[-3],4+c[-4],5+c[-3],5+c[-4]],
           X[4+c[-2],5+c[-4],5+c[-2],6+c[-4]],
           X[5+c[-2],5+c[-3],6+c[-2],6+c[-3]]}]]]]]]]],
 {2,3,4,5,6,1,7,8},{-1,-2,
  -2,
  -2,
  -2,
  0,
  0}}
```

In[•]:=
```
Knot2Bandtangle[PD@Knot[8, 20]]
```

Out[•]=
```
{{G1[3, 4], G1[7], G-1[2, 3, 4, 5, 6], G-1[1, 2, 3, 4, 5, 6]}, {1-1, 1-1, 21, 31, 31, 41, 41, 21}}
```

In[•]:=
```
Matching2Crossings[M_] := Module[{counters = ConstantArray[0, Length[M] / 2],
    WM, cur, pcur2, curcount, pair = 0, pd = {}},
   (*First rename the elements of M so that they are all in 1,
   2,..LengthM/2, output is with negative strand names.*)
   counters;
   WM = M /. Thread[DeleteDuplicates[M] → Range[Length[M] / 2]];
   While[pair < 100 && Length[WM] > 0, (*WM is a copy of M
     from which we keep removing pairs, starting with the leftmost*)
    pair++;
    (*Print["pair= ",pair];*)
    cur = WM[[1]]; (*The current symbol to work with*)
    (*Print["cur= ",cur];*)
    pcur2 = Position[WM, cur][[2, 1]];
    (*Print["pcur2= ",pcur2];*)
    curcount = 0;
    Table[
     (*Print["i= ",i," WM[i]= ",WM[[i]];*)
     If[Position[WM, WM[[i]]][[2, 1]] > pcur2,
       pd = Append[pd, X[c[-WM[[i]]] + counters[[WM[[i]]] - 1,
          c[-cur] + curcount + 1, c[-WM[[i]]] + counters[[WM[[i]]], c[-cur] + curcount]];
       counters[[WM[[i]]]--; curcount++;
      ];

     , {i, 2, pcur2 - 1}];
    WM = DeleteCases[WM, cur];
   ];
   (*Shift the crossing labels so that
      each component c[i] starts with c[i]+1 and then c[i]+2...*)

   pd = pd /. Table[
      Module[{cas}, cas = Cases[pd /. {X → List} // Flatten // Sort, c[i] + u_] /. c[i] → 0;
       If[Length[cas] > 0, c[i] → c[i] - If[First@cas < 0, First@cas, 0], Nothing]
      ],
      {i, -1, -Length[M] / 2, -1}
     ];
   pd /. Table[c[i] → c[i] + 1, {i, -1, -Length[M] / 2, -1}]
  ];
Matching2Crossings[{a, f, e, b, c, f, c, b, d, e, a, d}]
```

Out[•]=
```
{X[2 + c[-6], 2 + c[-1], 3 + c[-6], 1 + c[-1]],
 X[1 + c[-3], 2 + c[-2], 2 + c[-3], 1 + c[-2]], X[1 + c[-4], 3 + c[-2], 2 + c[-4], 2 + c[-2]],
 X[1 + c[-5], 4 + c[-2], 2 + c[-5], 3 + c[-2]], X[1 + c[-6], 3 + c[-3], 2 + c[-6], 2 + c[-3]]}
```

In[•]:=
```
Matching2Crossings[{a, b, a, b}]
```

Out[•]=
```
{X[1 + c[-2], 2 + c[-1], 2 + c[-2], 1 + c[-1]]}
```

In[•]:=
```
LenComp_comp_[pd_] :=
 Module[{Cas}, Cas = Cases[pd /. {X → List} // Flatten, c[comp] + u_] /. c[comp] → 0;
  If[Length[Cas] > 0, Max@Cas, 1]](*Empty components have length 1!*)
```

```
In[•]:= LenComp1@{X[2 + c[-6], 2 + c[-1], 3 + c[-6], 1 + c[-1]],
   X[1 + c[-3], 2 + c[-2], 2 + c[-3], 1 + c[-2]], X[1 + c[-4], 3 + c[-2], 2 + c[-4], 2 + c[-2]],
   X[1 + c[-5], 4 + c[-2], 2 + c[-5], 3 + c[-2]], X[1 + c[-6], 3 + c[-3], 2 + c[-6], 2 + c[-3]]}
Out[•]= 1

In[•]:= Flip_i_[X[p_, q_, r_, s_]] :=
   Module[{}, If[Coefficient[p, c[i]] == 1, X[r, s, p, q], X[p, q, r, s]]];
  (*To reverse we need to flip the order of the half edges
  around the crossing if we reverse the understrand,
careful: renumbering below flips p and r again in the underpass case.
    We call the reversal function S for antipode but note it is the involutory one!*)
S_comp_[PDin_] := Module[{lencomp, u, pd = PDin},
   (*Are kinks reversed properly?? Assuming strand is not closed!*)
   lencomp = LenComp_comp[pd];
   pd = pd /. {c[comp] + u_ → c[comp] + lencomp + 1 - u};
   Flip_comp /@ pd
  ]
S_1r[{X[c[5] + 1, 2 + c[3], 2 + c[5], 1 + c[3]],
   X[c[6] + 1, 3 + c[3], 2 + c[6], 2 + c[3]], X[3 + c[1 r], 4 + c[3], 4 + c[1 r], 3 + c[3]],
   X[2 + c[1 r], 2 + c[4], 3 + c[1 r], 1 + c[4]], X[1 + c[1 r], 2 + c[2], 2 + c[1 r], 1 + c[2]]}]
Out[•]= {X[1 + c[5], 2 + c[3], 2 + c[5], 1 + c[3]],
  X[1 + c[6], 3 + c[3], 2 + c[6], 2 + c[3]], X[1 + c[r], 3 + c[3], 2 + c[r], 4 + c[3]],
  X[2 + c[r], 1 + c[4], 3 + c[r], 2 + c[4]], X[3 + c[r], 1 + c[2], 4 + c[r], 2 + c[2]]}

In[•]:= Knot2Bandtangle[PD@Knot[7, 6]]
Out[•]= {{G-1[8], G1[3], G1[1, 2]}, {11, 41, 31, 2-1, 31, 41, 11, 2-1}}

In[•]:= Garside2Braid[G1[7, 8, 9]]
Out[•]= {X[1 + c[-7], 2 + c[-8], 2 + c[-7], 1 + c[-8]],
  X[2 + c[-7], 2 + c[-9], 3 + c[-7], 1 + c[-9]], X[2 + c[-8], 3 + c[-9], 3 + c[-8], 2 + c[-9]]}

In[•]:= (*Stitching is traditionally called m, Assuming strands are not closed!*)
Clear[m];
m_comp1_,comp2_->newname_[pdin_] := Module[{pd = pdin, lencomp1},
   (*Print["Stitching ",comp1," ",comp2];*)
   lencomp1 = LenComp_comp1[pd];
   pd = pd /. {c[comp2] → c[newname] + lencomp1 - 1};
   pd = pd /. c[comp1] → c[newname]
  ]
```

In[ ]:=

```
ConstructExplicitTangle[{Garsides_, Matchings_}] :=
 Module[{L = Length[Matchings], bottomline, pd, curmatchpos1,
   curmatchpos2, matchcomp, Wbot, safety = 0, curS, halftwists, bottomends},
  {pd, bottomline, halftwists} = Garsides2Braid[Garsides, L];
  Wbot = bottomline;
  bottomends = bottomline;
  pd = Join[pd, Matching2Crossings[Matchings]];
  While[safety < 100 && Length[Wbot] > 0,
   safety++;
   curS = First@Wbot;
   pd = S_curS[pd];
   halftwists[[curS]] = halftwists[[curS]] + Matchings[[curS]][[2]]; halftwists;
   {{curmatchpos1}, {curmatchpos2}} = Position[Matchings, Matchings[[curS]]];
   matchcomp = -Length[Union@Matchings[[1 ;; curmatchpos1]]];
   If[curmatchpos1 == curS,
    (*Print["No reverse"];*)
    pd = m_{curS,matchcomp->curS}[pd];
    pd = m_{curS,curmatchpos2->curS}[pd];
    bottomends = bottomends /. curmatchpos2 → curmatchpos1;
    ,
    (*Print["Reverse"];*)
    pd = S_matchcomp[pd];
    pd = m_{curS,matchcomp->curS}[pd];
    pd = m_{curS,curmatchpos1->curS}[pd];
    bottomends = bottomends /. curmatchpos1 → curmatchpos2;
   ];
   (*turn halftwists into kinks and add them*)
   If[halftwists[[curmatchpos1]] + halftwists[[curmatchpos2]] > 0,
    pd = Join[pd,
       Table[X[c[1/2] + 2 + 2 i, c[1/2] + 2 + 2 i, c[1/2] + 3 + 2 i, c[1/2] + 1 + 2 i]
        , {i, 0, Abs[halftwists[[curmatchpos1]] + halftwists[[curmatchpos2]]] / 2 - 1}]];,
    pd = Join[pd,
       Table[X[c[1/2] + 1 + 2 i, c[1/2] + 2 + 2 i, c[1/2] + 2 + 2 i, c[1/2] + 3 + 2 i]
        , {i, 0, Abs[halftwists[[curmatchpos1]] + halftwists[[curmatchpos2]]] / 2 - 1}]];
   ];
   pd = m_{curS,1/2->curS}[pd];
   Wbot = DeleteCases[Wbot, curmatchpos1 | curmatchpos2];
  ];
  {pd, bottomends}
 ]
ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[8, 20]]
% /. X → List // Flatten // Sort
% // BinCounts
```

KnotTheory: Loading precomputed data in PD4Knots`.

Out[•]= {{X[10 + c[3], 11 + c[4], 11 + c[3], 10 + c[4]], X[9 + c[4], 9 + c[2], 10 + c[4], 10 + c[2]],
X[9 + c[3], 8 + c[2], 10 + c[3], 9 + c[2]], X[11 + c[4], 8 + c[2], 12 + c[4], 7 + c[2]],
X[9 + c[6], 6 + c[2], 10 + c[6], 7 + c[2]], X[8 + c[3], 8 + c[4], 9 + c[3], 9 + c[4]],
X[12 + c[4], 8 + c[4], 13 + c[4], 7 + c[4]], X[8 + c[6], 6 + c[4], 9 + c[6], 7 + c[4]],
X[13 + c[4], 8 + c[3], 14 + c[4], 7 + c[3]], X[7 + c[6], 6 + c[3], 8 + c[6], 7 + c[3]],
X[6 + c[6], 15 + c[4], 7 + c[6], 14 + c[4]], X[5 + c[6], 11 + c[2], 6 + c[6], 10 + c[2]],
X[15 + c[4], 11 + c[2], 16 + c[4], 12 + c[2]], X[5 + c[3], 13 + c[2], 6 + c[3], 12 + c[2]],
X[5 + c[4], 14 + c[2], 6 + c[4], 13 + c[2]], X[5 + c[2], 15 + c[2], 6 + c[2], 14 + c[2]],
X[16 + c[4], 5 + c[6], 17 + c[4], 4 + c[6]], X[4 + c[3], 3 + c[6], 5 + c[3], 4 + c[6]],
X[4 + c[4], 2 + c[6], 5 + c[4], 3 + c[6]], X[4 + c[2], 1 + c[6], 5 + c[2], 2 + c[6]],
X[3 + c[3], 18 + c[4], 4 + c[3], 17 + c[4]], X[3 + c[4], 19 + c[4], 4 + c[4], 18 + c[4]],
X[3 + c[2], 20 + c[4], 4 + c[2], 19 + c[4]], X[2 + c[4], 2 + c[3], 3 + c[4], 3 + c[3]],
X[2 + c[2], 1 + c[3], 3 + c[2], 2 + c[3]], X[1 + c[2], 1 + c[4], 2 + c[2], 2 + c[4]],
X[15 + c[2], 16 + c[2], 16 + c[2], 17 + c[2]], X[17 + c[2], 18 + c[2], 18 + c[2], 19 + c[2]],
X[20 + c[4], 21 + c[4], 21 + c[4], 22 + c[4]]}, {2, 4, 3, 4, 6, 2, 6, 3}}

Out[•]= {2, 2, 3, 3, 4, 4, 6, 6, 1 + c[2], 2 + c[2], 2 + c[2], 3 + c[2], 3 + c[2], 4 + c[2],
4 + c[2], 5 + c[2], 5 + c[2], 6 + c[2], 6 + c[2], 7 + c[2], 7 + c[2], 8 + c[2], 8 + c[2],
9 + c[2], 9 + c[2], 10 + c[2], 10 + c[2], 11 + c[2], 11 + c[2], 12 + c[2], 12 + c[2],
13 + c[2], 13 + c[2], 14 + c[2], 14 + c[2], 15 + c[2], 15 + c[2], 16 + c[2], 16 + c[2],
17 + c[2], 17 + c[2], 18 + c[2], 18 + c[2], 19 + c[2], 1 + c[3], 2 + c[3], 2 + c[3],
3 + c[3], 3 + c[3], 4 + c[3], 4 + c[3], 5 + c[3], 5 + c[3], 6 + c[3], 6 + c[3], 7 + c[3],
7 + c[3], 8 + c[3], 8 + c[3], 9 + c[3], 9 + c[3], 10 + c[3], 10 + c[3], 11 + c[3],
1 + c[4], 2 + c[4], 2 + c[4], 3 + c[4], 3 + c[4], 4 + c[4], 4 + c[4], 5 + c[4], 5 + c[4],
6 + c[4], 6 + c[4], 7 + c[4], 7 + c[4], 8 + c[4], 8 + c[4], 9 + c[4], 9 + c[4], 10 + c[4],
10 + c[4], 11 + c[4], 11 + c[4], 12 + c[4], 12 + c[4], 13 + c[4], 13 + c[4], 14 + c[4],
14 + c[4], 15 + c[4], 15 + c[4], 16 + c[4], 16 + c[4], 17 + c[4], 17 + c[4], 18 + c[4],
18 + c[4], 19 + c[4], 19 + c[4], 20 + c[4], 20 + c[4], 21 + c[4], 21 + c[4], 22 + c[4],
1 + c[6], 2 + c[6], 2 + c[6], 3 + c[6], 3 + c[6], 4 + c[6], 4 + c[6], 5 + c[6], 5 + c[6],
6 + c[6], 6 + c[6], 7 + c[6], 7 + c[6], 8 + c[6], 8 + c[6], 9 + c[6], 9 + c[6], 10 + c[6]}

Out[•]= {0, 2, 2, 2, 0, 2}

In[•]:= **Position[{X[1 + c[3], 2 + c[4], 2 + c[3], 1 + c[4]],**
X[2 + c[4], 1 + c[2], 3 + c[4], 2 + c[2]], X[2 + c[3], 2 + c[2], 3 + c[3], 3 + c[2]],
X[1 + c[5], 3 + c[2], 2 + c[5], 4 + c[2]], X[1 + c[6], 4 + c[2], 2 + c[6], 5 + c[2]],
X[3 + c[3], 3 + c[4], 4 + c[3], 4 + c[4]], X[2 + c[5], 4 + c[4], 3 + c[5], 5 + c[4]],
X[2 + c[6], 5 + c[4], 3 + c[6], 6 + c[4]], X[3 + c[5], 4 + c[3], 4 + c[5], 5 + c[3]],
X[3 + c[6], 5 + c[3], 4 + c[6], 6 + c[3]], X[4 + c[6], 4 + c[5], 5 + c[6], 5 + c[5]],
X[5 + c[6], 1 + c[1], 6 + c[6], 2 + c[1]], X[6 + c[5], 2 + c[1], 7 + c[5], 3 + c[1]],
X[6 + c[3], 3 + c[1], 7 + c[3], 4 + c[1]], X[5 + c[4], 4 + c[1], 6 + c[4], 5 + c[1]],
X[5 + c[2], 5 + c[1], 6 + c[2], 6 + c[1]], X[7 + c[5], 6 + c[6], 8 + c[5], 7 + c[6]],
X[7 + c[3], 7 + c[6], 8 + c[3], 8 + c[6]], X[6 + c[4], 8 + c[6], 7 + c[4], 9 + c[6]],
X[6 + c[2], 9 + c[6], 7 + c[2], 10 + c[6]], X[8 + c[3], 8 + c[5], 9 + c[3], 9 + c[5]],
X[7 + c[4], 9 + c[5], 8 + c[4], 10 + c[5]], X[7 + c[2], 10 + c[5], 8 + c[2], 11 + c[5]],
X[8 + c[4], 9 + c[3], 9 + c[4], 10 + c[3]], X[8 + c[2], 10 + c[3], 9 + c[2], 11 + c[3]],
X[9 + c[2], 9 + c[4], 10 + c[2], 10 + c[4]]}, 5 + c[5]]

Out[•]= {{11, 4}}

```
In[•]:= Position[{X[1 + c[3], 2 + c[4], 2 + c[3], 1 + c[4]],
    X[2 + c[4], 1 + c[2], 3 + c[4], 2 + c[2]], X[2 + c[3], 2 + c[2], 3 + c[3], 3 + c[2]],
    X[1 + c[5], 3 + c[2], 2 + c[5], 4 + c[2]], X[1 + c[6], 4 + c[2], 2 + c[6], 5 + c[2]],
    X[3 + c[3], 3 + c[4], 4 + c[3], 4 + c[4]], X[2 + c[5], 4 + c[4], 3 + c[5], 5 + c[4]],
    X[2 + c[6], 5 + c[4], 3 + c[6], 6 + c[4]], X[3 + c[5], 4 + c[3], 4 + c[5], 5 + c[3]],
    X[3 + c[6], 5 + c[3], 4 + c[6], 6 + c[3]], X[4 + c[6], 4 + c[5], 5 + c[6], 5 + c[5]],
    X[5 + c[6], 1 + c[1], 6 + c[6], 2 + c[1]], X[6 + c[5], 2 + c[1], 7 + c[5], 3 + c[1]],
    X[6 + c[3], 3 + c[1], 7 + c[3], 4 + c[1]], X[5 + c[4], 4 + c[1], 6 + c[4], 5 + c[1]],
    X[5 + c[2], 5 + c[1], 6 + c[2], 6 + c[1]], X[7 + c[5], 6 + c[6], 8 + c[5], 7 + c[6]],
    X[7 + c[3], 7 + c[6], 8 + c[3], 8 + c[6]], X[6 + c[4], 8 + c[6], 7 + c[4], 9 + c[6]],
    X[6 + c[2], 9 + c[6], 7 + c[2], 10 + c[6]], X[8 + c[3], 8 + c[5], 9 + c[3], 9 + c[5]],
    X[7 + c[4], 9 + c[5], 8 + c[4], 10 + c[5]], X[7 + c[2], 10 + c[5], 8 + c[2], 11 + c[5]],
    X[8 + c[4], 9 + c[3], 9 + c[4], 10 + c[3]], X[8 + c[2], 10 + c[3], 9 + c[2], 11 + c[3]],
    X[9 + c[2], 9 + c[4], 10 + c[2], 10 + c[4]]}, 5 + c[5]]
Out[•]= {{11, 4}}

In[•]:= BinCounts@{2, 2, 3, 3, 4, 4, 6, 6, 1 + c[2], 2 + c[2], 2 + c[2], 3 + c[2], 3 + c[2], 4 + c[2],
    4 + c[2], 5 + c[2], 5 + c[2], 6 + c[2], 6 + c[2], 7 + c[2], 7 + c[2], 8 + c[2], 8 + c[2],
    9 + c[2], 9 + c[2], 10 + c[2], 10 + c[2], 11 + c[2], 11 + c[2], 12 + c[2], 12 + c[2],
    13 + c[2], 13 + c[2], 14 + c[2], 14 + c[2], 15 + c[2], 15 + c[2], 16 + c[2], 16 + c[2],
    17 + c[2], 17 + c[2], 18 + c[2], 18 + c[2], 19 + c[2], 1 + c[3], 2 + c[3], 2 + c[3],
    3 + c[3], 3 + c[3], 4 + c[3], 4 + c[3], 5 + c[3], 5 + c[3], 6 + c[3], 6 + c[3], 7 + c[3],
    7 + c[3], 8 + c[3], 8 + c[3], 9 + c[3], 9 + c[3], 10 + c[3], 10 + c[3], 11 + c[3],
    1 + c[4], 2 + c[4], 2 + c[4], 3 + c[4], 3 + c[4], 4 + c[4], 4 + c[4], 5 + c[4], 5 + c[4],
    5 + c[4], 6 + c[4], 6 + c[4], 6 + c[4], 7 + c[4], 7 + c[4], 8 + c[4], 8 + c[4], 9 + c[4],
    9 + c[4], 10 + c[4], 10 + c[4], 11 + c[4], 11 + c[4], 12 + c[4], 12 + c[4], 13 + c[4],
    13 + c[4], 14 + c[4], 15 + c[4], 16 + c[4], 16 + c[4], 17 + c[4], 17 + c[4], 18 + c[4],
    18 + c[4], 19 + c[4], 19 + c[4], 20 + c[4], 20 + c[4], 21 + c[4], 21 + c[4], 22 + c[4],
    1 + c[6], 2 + c[6], 2 + c[6], 3 + c[6], 3 + c[6], 4 + c[6], 4 + c[6], 5 + c[6], 5 + c[6],
    6 + c[6], 6 + c[6], 7 + c[6], 7 + c[6], 8 + c[6], 8 + c[6], 9 + c[6], 9 + c[6], 10 + c[6]}
Out[•]= {0, 2, 2, 2, 0, 2}

In[•]:= {X[1 + c[1], 2 + c[2], 2 + c[1], 1 + c[2]],
     X[2 + c[4], 3 + c[1], 3 + c[4], 2 + c[1]], X[1 + c[4], 3 + c[2], 2 + c[4], 2 + c[2]],
     X[c[4], 1 + c[3], 1 + c[4], c[3]], X[4 + c[2], 4 + c[2], 5 + c[2], 3 + c[2]],
     X[4 + c[1], 4 + c[1], 5 + c[1], 3 + c[1]], X[2 + c[3], 2 + c[3], 3 + c[3], 1 + c[3]],
     X[3 + c[4], 4 + c[4], 4 + c[4], 5 + c[4]]} /. X → List // Flatten // Sort
Out[•]= {1 + c[1], 2 + c[1], 2 + c[1], 3 + c[1], 3 + c[1], 4 + c[1], 4 + c[1],
  5 + c[1], 1 + c[2], 2 + c[2], 2 + c[2], 3 + c[2], 3 + c[2], 4 + c[2], 4 + c[2],
  5 + c[2], c[3], 1 + c[3], 1 + c[3], 2 + c[3], 2 + c[3], 3 + c[3], c[4], 1 + c[4],
  1 + c[4], 2 + c[4], 2 + c[4], 3 + c[4], 3 + c[4], 4 + c[4], 4 + c[4], 5 + c[4]}
```

In[ ]:=
```
(*DoubleAll doubles all strand of a tangle,
it uses the auxilliary function Reparametrize*)
Reparametrize[pd_] := Module[{wpd = pd /. X → List // Flatten, comps, rules, len, ans},
  comps = wpd /. c[u_] + v_ → c[u] // Union;
  rules = Flatten@Table[
     len = Length[Cases[wpd, co + u_] // Union];
     Thread[(Cases[wpd, co + u_] // Union) → Table[co + i, {i, len}]]
     , {co, comps}];
  pd /. rules
  ]
DoubleAll[pd_] :=
 Reparametrize[Flatten[pd /. {X[p_, q_, r_, s_] → If[PositiveQ@X[p, q, r, s],
        {X[pL, (s + 1/2)R, (p + 1/2)L, sR], X[pR, qR, (p + 1/2)R, (s + 1/2)R],
         X[(p + 1/2)L, (s + 1/2)L, rL, sL], X[(p + 1/2)R, qL, rR, (s + 1/2)L]},
        {X[pR, qL, (p + 1/2)R, (q + 1/2)L], X[(p + 1/2)R, qR, rR, (q + 1/2)R],
         X[pL, (q + 1/2)L, (p + 1/2)L, sL], X[(p + 1/2)L, (q + 1/2)R, rL, sR]}
       ]}, 1] /. (c[u_] + w_)v_ → c[u v] + w]
(ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[8, 20]])[[1]] // DoubleAll
% /. X → List // Flatten // Sort
```

Out[ ]=
```
{X[19 + c[3 L], 20 + c[4 R], 20 + c[3 L], 19 + c[4 R]],
 X[19 + c[3 R], 21 + c[4 R], 20 + c[3 R], 20 + c[4 R]],
 X[20 + c[3 L], 20 + c[4 L], 21 + c[3 L], 19 + c[4 L]],
 X[20 + c[3 R], 21 + c[4 L], 21 + c[3 R], 20 + c[4 L]],
 X[17 + c[4 R], 17 + c[2 L], 18 + c[4 R], 18 + c[2 L]],
 X[18 + c[4 R], 17 + c[2 R], 19 + c[4 R], 18 + c[2 R]],
 X[17 + c[4 L], 18 + c[2 L], 18 + c[4 L], 19 + c[2 L]],
 X[18 + c[4 L], 18 + c[2 R], 19 + c[4 L], 19 + c[2 R]],
 X[17 + c[3 R], 15 + c[2 L], 18 + c[3 R], 16 + c[2 L]],
 X[18 + c[3 R], 15 + c[2 R], 19 + c[3 R], 16 + c[2 R]],
 X[17 + c[3 L], 16 + c[2 L], 18 + c[3 L], 17 + c[2 L]],
 X[18 + c[3 L], 16 + c[2 R], 19 + c[3 L], 17 + c[2 R]],
 X[21 + c[4 L], 14 + c[2 R], 22 + c[4 L], 13 + c[2 R]],
 X[21 + c[4 R], 15 + c[2 R], 22 + c[4 R], 14 + c[2 R]],
 X[22 + c[4 L], 14 + c[2 L], 23 + c[4 L], 13 + c[2 L]],
 X[22 + c[4 R], 15 + c[2 L], 23 + c[4 R], 14 + c[2 L]],
 X[17 + c[6 R], 11 + c[2 L], 18 + c[6 R], 12 + c[2 L]],
 X[18 + c[6 R], 11 + c[2 R], 19 + c[6 R], 12 + c[2 R]],
 X[17 + c[6 L], 12 + c[2 L], 18 + c[6 L], 13 + c[2 L]],
 X[18 + c[6 L], 12 + c[2 R], 19 + c[6 L], 13 + c[2 R]],
 X[15 + c[3 R], 15 + c[4 L], 16 + c[3 R], 16 + c[4 L]],
 X[16 + c[3 R], 15 + c[4 R], 17 + c[3 R], 16 + c[4 R]],
 X[15 + c[3 L], 16 + c[4 L], 16 + c[3 L], 17 + c[4 L]],
 X[16 + c[3 L], 16 + c[4 R], 17 + c[3 L], 17 + c[4 R]],
 X[23 + c[4 L], 14 + c[4 R], 24 + c[4 L], 13 + c[4 R]],
```

```
X[23 + c[4 R], 15 + c[4 R], 24 + c[4 R], 14 + c[4 R]],
X[24 + c[4 L], 14 + c[4 L], 25 + c[4 L], 13 + c[4 L]],
X[24 + c[4 R], 15 + c[4 L], 25 + c[4 R], 14 + c[4 L]],
X[15 + c[6 R], 11 + c[4 L], 16 + c[6 R], 12 + c[4 L]],
X[16 + c[6 R], 11 + c[4 R], 17 + c[6 R], 12 + c[4 R]],
X[15 + c[6 L], 12 + c[4 L], 16 + c[6 L], 13 + c[4 L]],
X[16 + c[6 L], 12 + c[4 R], 17 + c[6 L], 13 + c[4 R]],
X[25 + c[4 L], 14 + c[3 R], 26 + c[4 L], 13 + c[3 R]],
X[25 + c[4 R], 15 + c[3 R], 26 + c[4 R], 14 + c[3 R]],
X[26 + c[4 L], 14 + c[3 L], 27 + c[4 L], 13 + c[3 L]],
X[26 + c[4 R], 15 + c[3 L], 27 + c[4 R], 14 + c[3 L]],
X[13 + c[6 R], 11 + c[3 L], 14 + c[6 R], 12 + c[3 L]],
X[14 + c[6 R], 11 + c[3 R], 15 + c[6 R], 12 + c[3 R]],
X[13 + c[6 L], 12 + c[3 L], 14 + c[6 L], 13 + c[3 L]],
X[14 + c[6 L], 12 + c[3 R], 15 + c[6 L], 13 + c[3 R]],
X[11 + c[6 L], 28 + c[4 R], 12 + c[6 L], 27 + c[4 R]],
X[11 + c[6 R], 29 + c[4 R], 12 + c[6 R], 28 + c[4 R]],
X[12 + c[6 L], 28 + c[4 L], 13 + c[6 L], 27 + c[4 L]],
X[12 + c[6 R], 29 + c[4 L], 13 + c[6 R], 28 + c[4 L]],
X[9 + c[6 L], 20 + c[2 R], 10 + c[6 L], 19 + c[2 R]],
X[9 + c[6 R], 21 + c[2 R], 10 + c[6 R], 20 + c[2 R]],
X[10 + c[6 L], 20 + c[2 L], 11 + c[6 L], 19 + c[2 L]],
X[10 + c[6 R], 21 + c[2 L], 11 + c[6 R], 20 + c[2 L]],
X[29 + c[4 R], 21 + c[2 L], 30 + c[4 R], 22 + c[2 L]],
X[30 + c[4 R], 21 + c[2 R], 31 + c[4 R], 22 + c[2 R]],
X[29 + c[4 L], 22 + c[2 L], 30 + c[4 L], 23 + c[2 L]],
X[30 + c[4 L], 22 + c[2 R], 31 + c[4 L], 23 + c[2 R]],
X[9 + c[3 L], 24 + c[2 R], 10 + c[3 L], 23 + c[2 R]],
X[9 + c[3 R], 25 + c[2 R], 10 + c[3 R], 24 + c[2 R]],
X[10 + c[3 L], 24 + c[2 L], 11 + c[3 L], 23 + c[2 L]],
X[10 + c[3 R], 25 + c[2 L], 11 + c[3 R], 24 + c[2 L]],
X[9 + c[4 L], 26 + c[2 R], 10 + c[4 L], 25 + c[2 R]],
X[9 + c[4 R], 27 + c[2 R], 10 + c[4 R], 26 + c[2 R]],
X[10 + c[4 L], 26 + c[2 L], 11 + c[4 L], 25 + c[2 L]],
X[10 + c[4 R], 27 + c[2 L], 11 + c[4 R], 26 + c[2 L]],
X[9 + c[2 L], 28 + c[2 R], 10 + c[2 L], 27 + c[2 R]],
X[9 + c[2 R], 29 + c[2 R], 10 + c[2 R], 28 + c[2 R]],
X[10 + c[2 L], 28 + c[2 L], 11 + c[2 L], 27 + c[2 L]],
X[10 + c[2 R], 29 + c[2 L], 11 + c[2 R], 28 + c[2 L]],
X[31 + c[4 L], 8 + c[6 R], 32 + c[4 L], 7 + c[6 R]],
X[31 + c[4 R], 9 + c[6 R], 32 + c[4 R], 8 + c[6 R]],
X[32 + c[4 L], 8 + c[6 L], 33 + c[4 L], 7 + c[6 L]],
X[32 + c[4 R], 9 + c[6 L], 33 + c[4 R], 8 + c[6 L]], X[7 + c[3 R], 5 + c[6 L], 8 + c[3 R], 6 + c[6 L]],
X[8 + c[3 R], 5 + c[6 R], 9 + c[3 R], 6 + c[6 R]], X[7 + c[3 L], 6 + c[6 L], 8 + c[3 L], 7 + c[6 L]],
X[8 + c[3 L], 6 + c[6 R], 9 + c[3 L], 7 + c[6 R]], X[7 + c[4 R], 3 + c[6 L], 8 + c[4 R], 4 + c[6 L]],
X[8 + c[4 R], 3 + c[6 R], 9 + c[4 R], 4 + c[6 R]], X[7 + c[4 L], 4 + c[6 L], 8 + c[4 L], 5 + c[6 L]],
X[8 + c[4 L], 4 + c[6 R], 9 + c[4 L], 5 + c[6 R]], X[7 + c[2 R], 1 + c[6 L], 8 + c[2 R], 2 + c[6 L]],
X[8 + c[2 R], 1 + c[6 R], 9 + c[2 R], 2 + c[6 R]], X[7 + c[2 L], 2 + c[6 L], 8 + c[2 L], 3 + c[6 L]],
X[8 + c[2 L], 2 + c[6 R], 9 + c[2 L], 3 + c[6 R]], X[5 + c[3 L], 34 + c[4 R], 6 + c[3 L], 33 + c[4 R]],
X[5 + c[3 R], 35 + c[4 R], 6 + c[3 R], 34 + c[4 R]],
X[6 + c[3 L], 34 + c[4 L], 7 + c[3 L], 33 + c[4 L]],
```

```
X[6 + c[3 R], 35 + c[4 L], 7 + c[3 R], 34 + c[4 L]],
X[5 + c[4 L], 36 + c[4 R], 6 + c[4 L], 35 + c[4 R]],
X[5 + c[4 R], 37 + c[4 R], 6 + c[4 R], 36 + c[4 R]],
X[6 + c[4 L], 36 + c[4 L], 7 + c[4 L], 35 + c[4 L]],
X[6 + c[4 R], 37 + c[4 L], 7 + c[4 R], 36 + c[4 L]],
X[5 + c[2 L], 38 + c[4 R], 6 + c[2 L], 37 + c[4 R]],
X[5 + c[2 R], 39 + c[4 R], 6 + c[2 R], 38 + c[4 R]],
X[6 + c[2 L], 38 + c[4 L], 7 + c[2 L], 37 + c[4 L]],
X[6 + c[2 R], 39 + c[4 L], 7 + c[2 R], 38 + c[4 L]],
X[3 + c[4 R], 3 + c[3 L], 4 + c[4 R], 4 + c[3 L]], X[4 + c[4 R], 3 + c[3 R], 5 + c[4 R], 4 + c[3 R]],
X[3 + c[4 L], 4 + c[3 L], 4 + c[4 L], 5 + c[3 L]], X[4 + c[4 L], 4 + c[3 R], 5 + c[4 L], 5 + c[3 R]],
X[3 + c[2 R], 1 + c[3 L], 4 + c[2 R], 2 + c[3 L]], X[4 + c[2 R], 1 + c[3 R], 5 + c[2 R], 2 + c[3 R]],
X[3 + c[2 L], 2 + c[3 L], 4 + c[2 L], 3 + c[3 L]], X[4 + c[2 L], 2 + c[3 R], 5 + c[2 L], 3 + c[3 R]],
X[1 + c[2 R], 1 + c[4 L], 2 + c[2 R], 2 + c[4 L]], X[2 + c[2 R], 1 + c[4 R], 3 + c[2 R], 2 + c[4 R]],
X[1 + c[2 L], 2 + c[4 L], 2 + c[2 L], 3 + c[4 L]], X[2 + c[2 L], 2 + c[4 R], 3 + c[2 L], 3 + c[4 R]],
X[29 + c[2 R], 31 + c[2 L], 30 + c[2 R], 32 + c[2 L]],
X[30 + c[2 R], 31 + c[2 R], 31 + c[2 R], 32 + c[2 R]],
X[29 + c[2 L], 32 + c[2 L], 30 + c[2 L], 33 + c[2 L]],
X[30 + c[2 L], 32 + c[2 R], 31 + c[2 L], 33 + c[2 R]],
X[33 + c[2 R], 35 + c[2 L], 34 + c[2 R], 36 + c[2 L]],
X[34 + c[2 R], 35 + c[2 R], 35 + c[2 R], 36 + c[2 R]],
X[33 + c[2 L], 36 + c[2 L], 34 + c[2 L], 37 + c[2 L]],
X[34 + c[2 L], 36 + c[2 R], 35 + c[2 L], 37 + c[2 R]],
X[39 + c[4 R], 41 + c[4 L], 40 + c[4 R], 42 + c[4 L]],
X[40 + c[4 R], 41 + c[4 R], 41 + c[4 R], 42 + c[4 R]],
X[39 + c[4 L], 42 + c[4 L], 40 + c[4 L], 43 + c[4 L]],
X[40 + c[4 L], 42 + c[4 R], 41 + c[4 L], 43 + c[4 R]]}
```

Out[•]=

```
{1 + c[2 L], 2 + c[2 L], 2 + c[2 L], 3 + c[2 L], 3 + c[2 L], 4 + c[2 L], 4 + c[2 L], 5 + c[2 L],
 5 + c[2 L], 6 + c[2 L], 6 + c[2 L], 7 + c[2 L], 7 + c[2 L], 8 + c[2 L], 8 + c[2 L], 9 + c[2 L],
 9 + c[2 L], 10 + c[2 L], 10 + c[2 L], 11 + c[2 L], 11 + c[2 L], 12 + c[2 L], 12 + c[2 L], 13 + c[2 L],
 13 + c[2 L], 14 + c[2 L], 14 + c[2 L], 15 + c[2 L], 15 + c[2 L], 16 + c[2 L], 16 + c[2 L],
 17 + c[2 L], 17 + c[2 L], 18 + c[2 L], 18 + c[2 L], 19 + c[2 L], 19 + c[2 L], 20 + c[2 L],
 20 + c[2 L], 21 + c[2 L], 21 + c[2 L], 22 + c[2 L], 22 + c[2 L], 23 + c[2 L], 23 + c[2 L],
 24 + c[2 L], 24 + c[2 L], 25 + c[2 L], 25 + c[2 L], 26 + c[2 L], 26 + c[2 L], 27 + c[2 L],
 27 + c[2 L], 28 + c[2 L], 28 + c[2 L], 29 + c[2 L], 29 + c[2 L], 30 + c[2 L], 30 + c[2 L],
 31 + c[2 L], 31 + c[2 L], 32 + c[2 L], 32 + c[2 L], 33 + c[2 L], 33 + c[2 L], 34 + c[2 L],
 34 + c[2 L], 35 + c[2 L], 35 + c[2 L], 36 + c[2 L], 36 + c[2 L], 37 + c[2 L], 1 + c[3 L], 2 + c[3 L],
 2 + c[3 L], 3 + c[3 L], 3 + c[3 L], 4 + c[3 L], 4 + c[3 L], 5 + c[3 L], 5 + c[3 L], 6 + c[3 L],
 6 + c[3 L], 7 + c[3 L], 7 + c[3 L], 8 + c[3 L], 8 + c[3 L], 9 + c[3 L], 9 + c[3 L], 10 + c[3 L],
 10 + c[3 L], 11 + c[3 L], 11 + c[3 L], 12 + c[3 L], 12 + c[3 L], 13 + c[3 L], 13 + c[3 L],
 14 + c[3 L], 14 + c[3 L], 15 + c[3 L], 15 + c[3 L], 16 + c[3 L], 16 + c[3 L], 17 + c[3 L],
 17 + c[3 L], 18 + c[3 L], 18 + c[3 L], 19 + c[3 L], 19 + c[3 L], 20 + c[3 L], 20 + c[3 L],
 21 + c[3 L], 1 + c[4 L], 2 + c[4 L], 2 + c[4 L], 3 + c[4 L], 3 + c[4 L], 4 + c[4 L], 4 + c[4 L],
 5 + c[4 L], 5 + c[4 L], 6 + c[4 L], 6 + c[4 L], 7 + c[4 L], 7 + c[4 L], 8 + c[4 L], 8 + c[4 L],
 9 + c[4 L], 9 + c[4 L], 10 + c[4 L], 10 + c[4 L], 11 + c[4 L], 11 + c[4 L], 12 + c[4 L], 12 + c[4 L],
 13 + c[4 L], 13 + c[4 L], 14 + c[4 L], 14 + c[4 L], 15 + c[4 L], 15 + c[4 L], 16 + c[4 L],
 16 + c[4 L], 17 + c[4 L], 17 + c[4 L], 18 + c[4 L], 18 + c[4 L], 19 + c[4 L], 19 + c[4 L],
 20 + c[4 L], 20 + c[4 L], 21 + c[4 L], 21 + c[4 L], 22 + c[4 L], 22 + c[4 L], 23 + c[4 L],
 23 + c[4 L], 24 + c[4 L], 24 + c[4 L], 25 + c[4 L], 25 + c[4 L], 26 + c[4 L], 26 + c[4 L],
 27 + c[4 L], 27 + c[4 L], 28 + c[4 L], 28 + c[4 L], 29 + c[4 L], 29 + c[4 L], 30 + c[4 L],
 30 + c[4 L], 31 + c[4 L], 31 + c[4 L], 32 + c[4 L], 32 + c[4 L], 33 + c[4 L], 33 + c[4 L],
```

```
34 + c[4 L], 34 + c[4 L], 35 + c[4 L], 35 + c[4 L], 36 + c[4 L], 36 + c[4 L], 37 + c[4 L],
37 + c[4 L], 38 + c[4 L], 38 + c[4 L], 39 + c[4 L], 39 + c[4 L], 40 + c[4 L], 40 + c[4 L],
41 + c[4 L], 41 + c[4 L], 42 + c[4 L], 42 + c[4 L], 43 + c[4 L], 1 + c[6 L], 2 + c[6 L], 2 + c[6 L],
3 + c[6 L], 3 + c[6 L], 4 + c[6 L], 4 + c[6 L], 5 + c[6 L], 5 + c[6 L], 6 + c[6 L], 6 + c[6 L],
7 + c[6 L], 7 + c[6 L], 8 + c[6 L], 8 + c[6 L], 9 + c[6 L], 9 + c[6 L], 10 + c[6 L], 10 + c[6 L],
11 + c[6 L], 11 + c[6 L], 12 + c[6 L], 12 + c[6 L], 13 + c[6 L], 13 + c[6 L], 14 + c[6 L],
14 + c[6 L], 15 + c[6 L], 15 + c[6 L], 16 + c[6 L], 16 + c[6 L], 17 + c[6 L], 17 + c[6 L],
18 + c[6 L], 18 + c[6 L], 19 + c[6 L], 1 + c[2 R], 2 + c[2 R], 2 + c[2 R], 3 + c[2 R], 3 + c[2 R],
4 + c[2 R], 4 + c[2 R], 5 + c[2 R], 5 + c[2 R], 6 + c[2 R], 6 + c[2 R], 7 + c[2 R], 7 + c[2 R],
8 + c[2 R], 8 + c[2 R], 9 + c[2 R], 9 + c[2 R], 10 + c[2 R], 10 + c[2 R], 11 + c[2 R], 11 + c[2 R],
12 + c[2 R], 12 + c[2 R], 13 + c[2 R], 13 + c[2 R], 14 + c[2 R], 14 + c[2 R], 15 + c[2 R],
15 + c[2 R], 16 + c[2 R], 16 + c[2 R], 17 + c[2 R], 17 + c[2 R], 18 + c[2 R], 18 + c[2 R],
19 + c[2 R], 19 + c[2 R], 20 + c[2 R], 20 + c[2 R], 21 + c[2 R], 21 + c[2 R], 22 + c[2 R],
22 + c[2 R], 23 + c[2 R], 23 + c[2 R], 24 + c[2 R], 24 + c[2 R], 25 + c[2 R], 25 + c[2 R],
26 + c[2 R], 26 + c[2 R], 27 + c[2 R], 27 + c[2 R], 28 + c[2 R], 28 + c[2 R], 29 + c[2 R],
29 + c[2 R], 30 + c[2 R], 30 + c[2 R], 31 + c[2 R], 31 + c[2 R], 32 + c[2 R], 32 + c[2 R],
33 + c[2 R], 33 + c[2 R], 34 + c[2 R], 34 + c[2 R], 35 + c[2 R], 35 + c[2 R], 36 + c[2 R],
36 + c[2 R], 37 + c[2 R], 1 + c[3 R], 2 + c[3 R], 2 + c[3 R], 3 + c[3 R], 3 + c[3 R], 4 + c[3 R],
4 + c[3 R], 5 + c[3 R], 5 + c[3 R], 6 + c[3 R], 6 + c[3 R], 7 + c[3 R], 7 + c[3 R], 8 + c[3 R],
8 + c[3 R], 9 + c[3 R], 9 + c[3 R], 10 + c[3 R], 10 + c[3 R], 11 + c[3 R], 11 + c[3 R], 12 + c[3 R],
12 + c[3 R], 13 + c[3 R], 13 + c[3 R], 14 + c[3 R], 14 + c[3 R], 15 + c[3 R], 15 + c[3 R],
16 + c[3 R], 16 + c[3 R], 17 + c[3 R], 17 + c[3 R], 18 + c[3 R], 18 + c[3 R], 19 + c[3 R],
19 + c[3 R], 20 + c[3 R], 20 + c[3 R], 21 + c[3 R], 1 + c[4 R], 2 + c[4 R], 2 + c[4 R], 3 + c[4 R],
3 + c[4 R], 4 + c[4 R], 4 + c[4 R], 5 + c[4 R], 5 + c[4 R], 6 + c[4 R], 6 + c[4 R], 7 + c[4 R],
7 + c[4 R], 8 + c[4 R], 8 + c[4 R], 9 + c[4 R], 9 + c[4 R], 10 + c[4 R], 10 + c[4 R], 11 + c[4 R],
11 + c[4 R], 12 + c[4 R], 12 + c[4 R], 13 + c[4 R], 13 + c[4 R], 14 + c[4 R], 14 + c[4 R],
15 + c[4 R], 15 + c[4 R], 16 + c[4 R], 16 + c[4 R], 17 + c[4 R], 17 + c[4 R], 18 + c[4 R],
18 + c[4 R], 19 + c[4 R], 19 + c[4 R], 20 + c[4 R], 20 + c[4 R], 21 + c[4 R], 21 + c[4 R],
22 + c[4 R], 22 + c[4 R], 23 + c[4 R], 23 + c[4 R], 24 + c[4 R], 24 + c[4 R], 25 + c[4 R],
25 + c[4 R], 26 + c[4 R], 26 + c[4 R], 27 + c[4 R], 27 + c[4 R], 28 + c[4 R], 28 + c[4 R],
29 + c[4 R], 29 + c[4 R], 30 + c[4 R], 30 + c[4 R], 31 + c[4 R], 31 + c[4 R], 32 + c[4 R],
32 + c[4 R], 33 + c[4 R], 33 + c[4 R], 34 + c[4 R], 34 + c[4 R], 35 + c[4 R], 35 + c[4 R],
36 + c[4 R], 36 + c[4 R], 37 + c[4 R], 37 + c[4 R], 38 + c[4 R], 38 + c[4 R], 39 + c[4 R],
39 + c[4 R], 40 + c[4 R], 40 + c[4 R], 41 + c[4 R], 41 + c[4 R], 42 + c[4 R], 42 + c[4 R],
43 + c[4 R], 1 + c[6 R], 2 + c[6 R], 2 + c[6 R], 3 + c[6 R], 3 + c[6 R], 4 + c[6 R], 4 + c[6 R],
5 + c[6 R], 5 + c[6 R], 6 + c[6 R], 6 + c[6 R], 7 + c[6 R], 7 + c[6 R], 8 + c[6 R], 8 + c[6 R],
9 + c[6 R], 9 + c[6 R], 10 + c[6 R], 10 + c[6 R], 11 + c[6 R], 11 + c[6 R], 12 + c[6 R],
12 + c[6 R], 13 + c[6 R], 13 + c[6 R], 14 + c[6 R], 14 + c[6 R], 15 + c[6 R], 15 + c[6 R],
16 + c[6 R], 16 + c[6 R], 17 + c[6 R], 17 + c[6 R], 18 + c[6 R], 18 + c[6 R], 19 + c[6 R]}
```

In[ ]:=

```
(*This turns the output of ConstructExplicitTangle into a pd code
 of a knot by thickening it in the standard way and then stitching
 in the order given by how the feet of the tangle meet the bottom.*)
ThickenTangle[pdTangle_, bottom_] :=
 Module[{pd = pdTangle, bot = bottom, strand, Li, Liold},
  (*Shift the names of the components by 1
    to avoid annoying bugs with 1R = R and 1L = L*)
  bot = bot + 1;
  pd = pd /. {c[u_] → c[u + 1]};
  strand = bot // DeleteDuplicates;
  pd = DoubleAll[pd];
  (*pd/.X→List//Flatten//Sort//Echo;*)
  For[i = 1, i <= Length[bot] / 2, i++,
   pd = pd // S_strand[[i]]*R
  ];
  (*pd/.X→List//Flatten//Sort//Echo;*)
  Li = Table[If[{j} == FirstPosition[bot, bot[[j]]],
     {L bot[[j]], bot[[j]] R}, {R bot[[j]], bot[[j]] L}], {j, 1, Length[bot]}];
  For[i = 1, i ≤ Length[Li] - 1, i++,
   pd = pd // m_Li[[i,2]],Li[[i+1,1]]->Li[[i+1,1]];
   Li = Li /. Li[[i, 2]] → Li[[i + 1, 1]];
  ];
  pd = pd /. LenComp_Li[[1,2]][pd] → 1; (*close the loop*)
  pd = pd /. {c[u_] → 0}
 ]
ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[8, 20]]
% /. X → List // Flatten // Sort
```

Out[•]= {X[231, 194, 232, 195], X[152, 194, 153, 193], X[232, 110, 1, 109], X[151, 110, 152, 111],
X[196, 18, 197, 17], X[195, 74, 196, 75], X[107, 18, 108, 19], X[108, 74, 109, 73],
X[154, 16, 155, 15], X[153, 76, 154, 77], X[229, 16, 230, 17], X[230, 76, 231, 75],
X[111, 78, 112, 79], X[192, 78, 193, 77], X[112, 14, 113, 13], X[191, 14, 192, 15],
X[38, 12, 39, 11], X[37, 80, 38, 81], X[149, 12, 150, 13], X[150, 80, 151, 79],
X[156, 106, 157, 105], X[155, 198, 156, 199], X[227, 106, 228, 107], X[228, 198, 229, 197],
X[113, 200, 114, 201], X[190, 200, 191, 199], X[114, 104, 115, 103], X[189, 104, 190, 105],
X[40, 102, 41, 101], X[39, 202, 40, 203], X[147, 102, 148, 103], X[148, 202, 149, 201],
X[115, 158, 116, 159], X[188, 158, 189, 157], X[116, 226, 117, 225], X[187, 226, 188, 227],
X[42, 224, 43, 223], X[41, 160, 42, 161], X[145, 224, 146, 225], X[146, 160, 147, 159],
X[143, 186, 144, 187], X[44, 186, 45, 185], X[144, 118, 145, 117], X[43, 118, 44, 119],
X[141, 72, 142, 73], X[46, 72, 47, 71], X[142, 20, 143, 19], X[45, 20, 46, 21],
X[184, 22, 185, 21], X[183, 70, 184, 71], X[119, 22, 120, 23], X[120, 70, 121, 69],
X[221, 68, 222, 69], X[162, 68, 163, 67], X[222, 24, 223, 23], X[161, 24, 162, 25],
X[99, 66, 100, 67], X[204, 66, 205, 65], X[100, 26, 101, 25], X[203, 26, 204, 27],
X[9, 64, 10, 65], X[82, 64, 83, 63], X[10, 28, 11, 27], X[81, 28, 82, 29],
X[121, 48, 122, 49], X[182, 48, 183, 47], X[122, 140, 123, 139], X[181, 140, 182, 141],
X[164, 138, 165, 137], X[163, 50, 164, 51], X[219, 138, 220, 139], X[220, 50, 221, 49],
X[206, 136, 207, 135], X[205, 52, 206, 53], X[97, 136, 98, 137], X[98, 52, 99, 51],
X[84, 134, 85, 133], X[83, 54, 84, 55], X[7, 134, 8, 135], X[8, 54, 9, 53],
X[217, 180, 218, 181], X[166, 180, 167, 179], X[218, 124, 219, 123], X[165, 124, 166, 125],
X[95, 178, 96, 179], X[208, 178, 209, 177], X[96, 126, 97, 125], X[207, 126, 208, 127],
X[5, 176, 6, 177], X[86, 176, 87, 175], X[6, 128, 7, 127], X[85, 128, 86, 129],
X[210, 216, 211, 215], X[209, 168, 210, 169], X[93, 216, 94, 217], X[94, 168, 95, 167],
X[88, 214, 89, 213], X[87, 170, 88, 171], X[3, 214, 4, 215], X[4, 170, 5, 169],
X[90, 92, 91, 91], X[89, 212, 90, 213], X[1, 92, 2, 93], X[2, 212, 3, 211],
X[62, 32, 63, 31], X[61, 60, 62, 61], X[29, 32, 30, 33], X[30, 60, 31, 59],
X[58, 36, 59, 35], X[57, 56, 58, 57], X[33, 36, 34, 37], X[34, 56, 35, 55],
X[174, 132, 175, 131], X[173, 172, 174, 173], X[129, 132, 130, 133], X[130, 172, 131, 171]}

```
Out[•]= {1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15,
  15, 16, 16, 17, 17, 18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 23, 24, 24, 25, 25, 26, 26,
  27, 27, 28, 28, 29, 29, 30, 30, 31, 31, 32, 32, 33, 33, 34, 34, 35, 35, 36, 36, 37, 37, 38,
  38, 39, 39, 40, 40, 41, 41, 42, 42, 43, 43, 44, 44, 45, 45, 46, 46, 47, 47, 48, 48, 49, 49,
  50, 50, 51, 51, 52, 52, 53, 53, 54, 54, 55, 55, 56, 56, 57, 57, 58, 58, 59, 59, 60, 60,
  61, 61, 62, 62, 63, 63, 64, 64, 65, 65, 66, 66, 67, 67, 68, 68, 69, 69, 70, 70, 71, 71,
  72, 72, 73, 73, 74, 74, 75, 75, 76, 76, 77, 77, 78, 78, 79, 79, 80, 80, 81, 81, 82, 82,
  83, 83, 84, 84, 85, 85, 86, 86, 87, 87, 88, 88, 89, 89, 90, 90, 91, 91, 92, 92, 93, 93,
  94, 94, 95, 95, 96, 96, 97, 97, 98, 98, 99, 99, 100, 100, 101, 101, 102, 102, 103, 103,
  104, 104, 105, 105, 106, 106, 107, 107, 108, 108, 109, 109, 110, 110, 111, 111, 112, 112,
  113, 113, 114, 114, 115, 115, 116, 116, 117, 117, 118, 118, 119, 119, 120, 120, 121, 121,
  122, 122, 123, 123, 124, 124, 125, 125, 126, 126, 127, 127, 128, 128, 129, 129, 130, 130,
  131, 131, 132, 132, 133, 133, 134, 134, 135, 135, 136, 136, 137, 137, 138, 138, 139,
  139, 140, 140, 141, 141, 142, 142, 143, 143, 144, 144, 145, 145, 146, 146, 147, 147,
  148, 148, 149, 149, 150, 150, 151, 151, 152, 152, 153, 153, 154, 154, 155, 155, 156,
  156, 157, 157, 158, 158, 159, 159, 160, 160, 161, 161, 162, 162, 163, 163, 164, 164,
  165, 165, 166, 166, 167, 167, 168, 168, 169, 169, 170, 170, 171, 171, 172, 172, 173,
  173, 174, 174, 175, 175, 176, 176, 177, 177, 178, 178, 179, 179, 180, 180, 181, 181,
  182, 182, 183, 183, 184, 184, 185, 185, 186, 186, 187, 187, 188, 188, 189, 189, 190,
  190, 191, 191, 192, 192, 193, 193, 194, 194, 195, 195, 196, 196, 197, 197, 198, 198,
  199, 199, 200, 200, 201, 201, 202, 202, 203, 203, 204, 204, 205, 205, 206, 206, 207,
  207, 208, 208, 209, 209, 210, 210, 211, 211, 212, 212, 213, 213, 214, 214, 215, 215,
  216, 216, 217, 217, 218, 218, 219, 219, 220, 220, 221, 221, 222, 222, 223, 223, 224,
  224, 225, 225, 226, 226, 227, 227, 228, 228, 229, 229, 230, 230, 231, 231, 232, 232}
```

```
In[•]:= BinCounts@%
```

```
Out[•]= {0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2}
```

```
In[•]:= os = {1 + c[3 L], 2 + c[3 L], 2 + c[3 L], 3 + c[3 L], 3 + c[3 L], 4 + c[3 L], 4 + c[3 L], 5 + c[3 L],
    5 + c[3 L], 6 + c[3 L], 6 + c[3 L], 7 + c[3 L], 7 + c[3 L], 8 + c[3 L], 8 + c[3 L], 9 + c[3 L],
    9 + c[3 L], 10 + c[3 L], 10 + c[3 L], 11 + c[3 L], 11 + c[3 L], 12 + c[3 L], 12 + c[3 L],
    13 + c[3 L], 13 + c[3 L], 14 + c[3 L], 14 + c[3 L], 15 + c[3 L], 15 + c[3 L], 16 + c[3 L],
    16 + c[3 L], 17 + c[3 L], 17 + c[3 L], 18 + c[3 L], 18 + c[3 L], 19 + c[3 L], 19 + c[3 L],
    20 + c[3 L], 20 + c[3 L], 21 + c[3 L], 21 + c[3 L], 22 + c[3 L], 22 + c[3 L], 23 + c[3 L],
    23 + c[3 L], 24 + c[3 L], 24 + c[3 L], 25 + c[3 L], 25 + c[3 L], 26 + c[3 L], 26 + c[3 L],
    27 + c[3 L], 27 + c[3 L], 28 + c[3 L], 28 + c[3 L], 29 + c[3 L], 29 + c[3 L], 30 + c[3 L],
    30 + c[3 L], 31 + c[3 L], 31 + c[3 L], 32 + c[3 L], 32 + c[3 L], 33 + c[3 L], 33 + c[3 L],
    34 + c[3 L], 34 + c[3 L], 35 + c[3 L], 35 + c[3 L], 36 + c[3 L], 36 + c[3 L], 37 + c[3 L],
    1 + c[4 L], 2 + c[4 L], 2 + c[4 L], 3 + c[4 L], 3 + c[4 L], 4 + c[4 L], 4 + c[4 L], 5 + c[4 L],
    5 + c[4 L], 6 + c[4 L], 6 + c[4 L], 7 + c[4 L], 7 + c[4 L], 8 + c[4 L], 8 + c[4 L], 9 + c[4 L],
    9 + c[4 L], 10 + c[4 L], 10 + c[4 L], 11 + c[4 L], 11 + c[4 L], 12 + c[4 L], 12 + c[4 L],
    13 + c[4 L], 13 + c[4 L], 14 + c[4 L], 14 + c[4 L], 15 + c[4 L], 15 + c[4 L], 16 + c[4 L],
    16 + c[4 L], 17 + c[4 L], 17 + c[4 L], 18 + c[4 L], 18 + c[4 L], 19 + c[4 L], 19 + c[4 L],
    20 + c[4 L], 20 + c[4 L], 21 + c[4 L], 1 + c[5 L], 2 + c[5 L], 2 + c[5 L], 3 + c[5 L],
```

```
3 + c[5 L], 4 + c[5 L], 4 + c[5 L], 5 + c[5 L], 5 + c[5 L], 6 + c[5 L], 6 + c[5 L], 7 + c[5 L],
7 + c[5 L], 8 + c[5 L], 8 + c[5 L], 9 + c[5 L], 9 + c[5 L], 9 + c[5 L], 10 + c[5 L], 10 + c[5 L],
10 + c[5 L], 10 + c[5 L], 11 + c[5 L], 11 + c[5 L], 11 + c[5 L], 12 + c[5 L], 12 + c[5 L],
13 + c[5 L], 13 + c[5 L], 14 + c[5 L], 14 + c[5 L], 15 + c[5 L], 15 + c[5 L], 16 + c[5 L],
16 + c[5 L], 17 + c[5 L], 17 + c[5 L], 18 + c[5 L], 18 + c[5 L], 19 + c[5 L], 19 + c[5 L],
20 + c[5 L], 20 + c[5 L], 21 + c[5 L], 21 + c[5 L], 22 + c[5 L], 22 + c[5 L], 23 + c[5 L],
23 + c[5 L], 24 + c[5 L], 24 + c[5 L], 25 + c[5 L], 25 + c[5 L], 26 + c[5 L], 26 + c[5 L],
27 + c[5 L], 28 + c[5 L], 29 + c[5 L], 29 + c[5 L], 30 + c[5 L], 30 + c[5 L], 31 + c[5 L],
31 + c[5 L], 32 + c[5 L], 32 + c[5 L], 33 + c[5 L], 33 + c[5 L], 34 + c[5 L], 34 + c[5 L],
35 + c[5 L], 35 + c[5 L], 36 + c[5 L], 36 + c[5 L], 37 + c[5 L], 37 + c[5 L], 38 + c[5 L],
38 + c[5 L], 39 + c[5 L], 39 + c[5 L], 40 + c[5 L], 40 + c[5 L], 41 + c[5 L], 41 + c[5 L],
42 + c[5 L], 1 + c[7 L], 2 + c[7 L], 2 + c[7 L], 3 + c[7 L], 3 + c[7 L], 4 + c[7 L], 4 + c[7 L],
5 + c[7 L], 5 + c[7 L], 6 + c[7 L], 6 + c[7 L], 7 + c[7 L], 7 + c[7 L], 8 + c[7 L], 8 + c[7 L],
9 + c[7 L], 9 + c[7 L], 10 + c[7 L], 10 + c[7 L], 11 + c[7 L], 11 + c[7 L], 12 + c[7 L],
12 + c[7 L], 13 + c[7 L], 13 + c[7 L], 14 + c[7 L], 14 + c[7 L], 15 + c[7 L], 15 + c[7 L],
16 + c[7 L], 16 + c[7 L], 17 + c[7 L], 17 + c[7 L], 18 + c[7 L], 18 + c[7 L], 19 + c[7 L],
1 + c[3 R], 2 + c[3 R], 2 + c[3 R], 3 + c[3 R], 3 + c[3 R], 4 + c[3 R], 4 + c[3 R], 5 + c[3 R],
5 + c[3 R], 6 + c[3 R], 6 + c[3 R], 7 + c[3 R], 7 + c[3 R], 8 + c[3 R], 8 + c[3 R], 9 + c[3 R],
9 + c[3 R], 10 + c[3 R], 10 + c[3 R], 11 + c[3 R], 11 + c[3 R], 12 + c[3 R], 12 + c[3 R],
13 + c[3 R], 13 + c[3 R], 14 + c[3 R], 14 + c[3 R], 15 + c[3 R], 15 + c[3 R], 16 + c[3 R],
16 + c[3 R], 17 + c[3 R], 17 + c[3 R], 18 + c[3 R], 18 + c[3 R], 19 + c[3 R], 19 + c[3 R],
20 + c[3 R], 20 + c[3 R], 21 + c[3 R], 21 + c[3 R], 22 + c[3 R], 22 + c[3 R], 23 + c[3 R],
23 + c[3 R], 24 + c[3 R], 24 + c[3 R], 25 + c[3 R], 25 + c[3 R], 26 + c[3 R], 26 + c[3 R],
27 + c[3 R], 27 + c[3 R], 28 + c[3 R], 28 + c[3 R], 29 + c[3 R], 29 + c[3 R], 30 + c[3 R],
30 + c[3 R], 31 + c[3 R], 31 + c[3 R], 32 + c[3 R], 32 + c[3 R], 33 + c[3 R], 33 + c[3 R],
34 + c[3 R], 34 + c[3 R], 35 + c[3 R], 35 + c[3 R], 36 + c[3 R], 36 + c[3 R], 37 + c[3 R],
1 + c[4 R], 2 + c[4 R], 2 + c[4 R], 3 + c[4 R], 3 + c[4 R], 4 + c[4 R], 4 + c[4 R], 5 + c[4 R],
5 + c[4 R], 6 + c[4 R], 6 + c[4 R], 7 + c[4 R], 7 + c[4 R], 8 + c[4 R], 8 + c[4 R], 9 + c[4 R],
9 + c[4 R], 10 + c[4 R], 10 + c[4 R], 11 + c[4 R], 11 + c[4 R], 12 + c[4 R], 12 + c[4 R],
13 + c[4 R], 13 + c[4 R], 14 + c[4 R], 14 + c[4 R], 15 + c[4 R], 15 + c[4 R], 16 + c[4 R],
16 + c[4 R], 17 + c[4 R], 17 + c[4 R], 18 + c[4 R], 18 + c[4 R], 19 + c[4 R], 19 + c[4 R],
20 + c[4 R], 20 + c[4 R], 21 + c[4 R], 1 + c[5 R], 2 + c[5 R], 2 + c[5 R], 3 + c[5 R], 3 + c[5 R],
4 + c[5 R], 4 + c[5 R], 5 + c[5 R], 5 + c[5 R], 6 + c[5 R], 6 + c[5 R], 7 + c[5 R], 7 + c[5 R],
8 + c[5 R], 8 + c[5 R], 9 + c[5 R], 9 + c[5 R], 10 + c[5 R], 10 + c[5 R], 11 + c[5 R],
11 + c[5 R], 12 + c[5 R], 12 + c[5 R], 13 + c[5 R], 13 + c[5 R], 14 + c[5 R], 14 + c[5 R],
15 + c[5 R], 16 + c[5 R], 17 + c[5 R], 17 + c[5 R], 18 + c[5 R], 18 + c[5 R], 19 + c[5 R],
19 + c[5 R], 20 + c[5 R], 20 + c[5 R], 21 + c[5 R], 21 + c[5 R], 22 + c[5 R], 22 + c[5 R],
23 + c[5 R], 23 + c[5 R], 24 + c[5 R], 24 + c[5 R], 25 + c[5 R], 25 + c[5 R], 26 + c[5 R],
26 + c[5 R], 27 + c[5 R], 27 + c[5 R], 28 + c[5 R], 28 + c[5 R], 29 + c[5 R], 29 + c[5 R],
30 + c[5 R], 30 + c[5 R], 31 + c[5 R], 31 + c[5 R], 32 + c[5 R], 32 + c[5 R], 32 + c[5 R],
33 + c[5 R], 33 + c[5 R], 33 + c[5 R], 33 + c[5 R], 34 + c[5 R], 34 + c[5 R], 34 + c[5 R],
35 + c[5 R], 35 + c[5 R], 36 + c[5 R], 36 + c[5 R], 37 + c[5 R], 37 + c[5 R], 38 + c[5 R],
38 + c[5 R], 39 + c[5 R], 39 + c[5 R], 40 + c[5 R], 40 + c[5 R], 41 + c[5 R], 41 + c[5 R],
42 + c[5 R], 1 + c[7 R], 2 + c[7 R], 2 + c[7 R], 3 + c[7 R], 3 + c[7 R], 4 + c[7 R], 4 + c[7 R],
5 + c[7 R], 5 + c[7 R], 6 + c[7 R], 6 + c[7 R], 7 + c[7 R], 7 + c[7 R], 8 + c[7 R], 8 + c[7 R],
9 + c[7 R], 9 + c[7 R], 10 + c[7 R], 10 + c[7 R], 11 + c[7 R], 11 + c[7 R], 12 + c[7 R],
12 + c[7 R], 13 + c[7 R], 13 + c[7 R], 14 + c[7 R], 14 + c[7 R], 15 + c[7 R], 15 + c[7 R],
16 + c[7 R], 16 + c[7 R], 17 + c[7 R], 17 + c[7 R], 18 + c[7 R], 18 + c[7 R], 19 + c[7 R]};
```

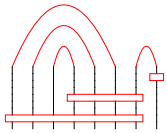
```
In[ ]:= (*Simple slow Kauffman bracket routine from KnotAtlas website,
fancy Jones doesnt work!*)
KB0[pd_] := Expand[
  Expand[Times @@ pd /. X[a_, b_, c_, d_] :> A P[a, d] P[b, c] + 1/A P[a, b] P[c, d]] //.
   {P[a_, b_] P[b_, c_] :> P[a, c], P[a_, b_]^2 :> P[a, a], P[a_, a_] -> -A^2 - 1/A^2}]
ShiftLaurent[P_] := P * A^-Exponent[P,A] / Coefficient[P, A^Exponent[P,A]] // Expand
KBshifted[pd_] := KB0[pd] // ShiftLaurent;
```

Testing

```
In[ ]:= Knot2Bandtangle[PD@Knot[8, 20]]
        DrawTangle@Knot2Bandtangle[PD@Knot[7, 3]]
```

```
Out[ ]= {{G1[3, 4], G1[7], G-1[2, 3, 4, 5, 6], G-1[1, 2, 3, 4, 5, 6]}, {1-1, 1-1, 21, 31, 31, 41, 41, 21}}
```

```
Out[ ]=
```



```
In[ ]:= ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[8, 20]]
```

```
Out[ ]= {{X[10 + c[3], 11 + c[4], 11 + c[3], 10 + c[4]], X[9 + c[4], 9 + c[2], 10 + c[4], 10 + c[2]],
  X[9 + c[3], 8 + c[2], 10 + c[3], 9 + c[2]], X[11 + c[4], 8 + c[2], 12 + c[4], 7 + c[2]],
  X[9 + c[6], 6 + c[2], 10 + c[6], 7 + c[2]], X[8 + c[3], 8 + c[4], 9 + c[3], 9 + c[4]],
  X[12 + c[4], 8 + c[4], 13 + c[4], 7 + c[4]], X[8 + c[6], 6 + c[4], 9 + c[6], 7 + c[4]],
  X[13 + c[4], 8 + c[3], 14 + c[4], 7 + c[3]], X[7 + c[6], 6 + c[3], 8 + c[6], 7 + c[3]],
  X[6 + c[6], 15 + c[4], 7 + c[6], 14 + c[4]], X[5 + c[6], 11 + c[2], 6 + c[6], 10 + c[2]],
  X[15 + c[4], 11 + c[2], 16 + c[4], 12 + c[2]], X[5 + c[3], 13 + c[2], 6 + c[3], 12 + c[2]],
  X[5 + c[4], 14 + c[2], 6 + c[4], 13 + c[2]], X[5 + c[2], 15 + c[2], 6 + c[2], 14 + c[2]],
  X[16 + c[4], 5 + c[6], 17 + c[4], 4 + c[6]], X[4 + c[3], 3 + c[6], 5 + c[3], 4 + c[6]],
  X[4 + c[4], 2 + c[6], 5 + c[4], 3 + c[6]], X[4 + c[2], 1 + c[6], 5 + c[2], 2 + c[6]],
  X[3 + c[3], 18 + c[4], 4 + c[3], 17 + c[4]], X[3 + c[4], 19 + c[4], 4 + c[4], 18 + c[4]],
  X[3 + c[2], 20 + c[4], 4 + c[2], 19 + c[4]], X[2 + c[4], 2 + c[3], 3 + c[4], 3 + c[3]],
  X[2 + c[2], 1 + c[3], 3 + c[2], 2 + c[3]], X[1 + c[2], 1 + c[4], 2 + c[2], 2 + c[4]],
  X[15 + c[2], 16 + c[2], 16 + c[2], 17 + c[2]], X[17 + c[2], 18 + c[2], 18 + c[2], 19 + c[2]],
  X[20 + c[4], 21 + c[4], 21 + c[4], 22 + c[4]]}, {2, 4, 3, 4, 6, 2, 6, 3}}
```

```
In[ ]:= ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[6, 3]]
```

```
Out[ ]= {X[55, 54, 56, 55], X[48, 54, 49, 53], X[56, 18, 57, 17], X[47, 18, 48, 19], X[30, 32, 31, 31],
 X[29, 16, 30, 17], X[1, 32, 2, 33], X[2, 16, 3, 15], X[59, 14, 60, 15], X[44, 14, 45, 13],
 X[60, 34, 61, 33], X[43, 34, 44, 35], X[57, 28, 58, 29], X[46, 28, 47, 27],
 X[58, 4, 59, 3], X[45, 4, 46, 5], X[26, 8, 27, 7], X[25, 24, 26, 25], X[5, 8, 6, 9],
 X[6, 24, 7, 23], X[12, 38, 13, 37], X[11, 10, 12, 11], X[35, 38, 36, 39], X[36, 10, 37, 9],
 X[21, 52, 22, 53], X[50, 52, 51, 51], X[22, 20, 23, 19], X[49, 20, 50, 21],
 X[63, 42, 64, 43], X[40, 42, 41, 41], X[64, 62, 1, 61], X[39, 62, 40, 63]}
```

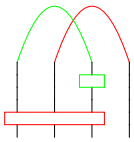
```
In[•]:= KBshifted[ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[3, 1]]] ==
         KBshifted[PD@Knot[3, 1]]
Out[•]= True
```

It works!!!! (for the trefoil)

KB0 is a bit slow for the 20 crossing diagram that comes out of 4_1.

```
In[•]:= Knot2Bandtangle[PD@Knot[4, 1]]
        DrawTangle@Knot2Bandtangle[PD@Knot[4, 1]]
Out[•]= {{G1[3], G-1[1, 2, 3]}, {21, 1-1, 21, 1-1}}
```

Out[•]=



```
In[•]:= ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[4, 1]]
Out[•]= {{X[2 + c[2], 5 + c[3], 3 + c[2], 4 + c[3]], X[2 + c[3], 6 + c[3], 3 + c[3], 5 + c[3]],
          X[1 + c[3], 1 + c[2], 2 + c[3], 2 + c[2]], X[3 + c[2], 3 + c[3], 4 + c[2], 4 + c[3]],
          X[4 + c[2], 5 + c[2], 5 + c[2], 6 + c[2]]}, {3, 2, 3, 2}}

In[•]:= ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[4, 1]]
Out[•]= {X[33, 24, 34, 25], X[18, 24, 19, 23], X[34, 8, 35, 7], X[17, 8, 18, 9], X[3, 22, 4, 23],
         X[28, 22, 29, 21], X[4, 10, 5, 9], X[27, 10, 28, 11], X[30, 32, 31, 31], X[29, 20, 30, 21],
         X[1, 32, 2, 33], X[2, 20, 3, 19], X[16, 6, 17, 5], X[15, 26, 16, 27], X[35, 6, 36, 7],
         X[36, 26, 37, 25], X[14, 40, 15, 39], X[13, 12, 14, 13], X[37, 40, 38, 1], X[38, 12, 39, 11]}

In[•]:= Timing[
         KBshifted[ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@Knot[4, 1]]] ==
          KBshifted[PD@Knot[4, 1]]]
Out[•]= {1127.86, True}

In[•]:= TestKnotJones[K_] :=
         Jones[Echo[PD @@ ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@K]]][
           t] == Jones[PD@K][t]
        TestKnotAlex[K_] :=
         Alexander[PD @@ ThickenTangle @@ ConstructExplicitTangle@Knot2Bandtangle[PD@K]][t] ==
          Alexander[PD@K][t]
```

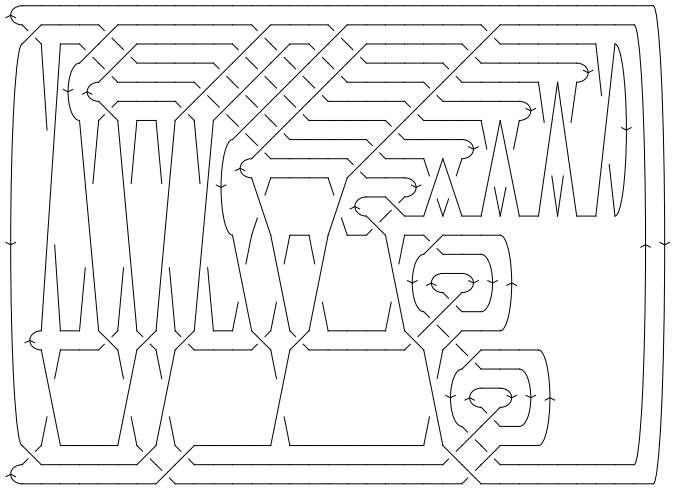
In[•]:=
```
TestKnotJones[Knot[7, 7]]
```

» PD[X[15, 130, 16, 131], X[72, 130, 73, 129], X[16, 54, 17, 53], X[71, 54, 72, 55], X[70, 152, 71, 151],
X[69, 112, 70, 113], X[17, 152, 18, 153], X[18, 112, 19, 111], X[68, 50, 69, 49],
X[67, 134, 68, 135], X[19, 50, 20, 51], X[20, 134, 21, 133], X[66, 96, 67, 95], X[65, 32, 66, 33],
X[21, 96, 22, 97], X[22, 32, 23, 31], X[64, 10, 65, 9], X[63, 78, 64, 79], X[23, 10, 24, 11],
X[24, 78, 25, 77], X[128, 150, 129, 149], X[127, 114, 128, 115], X[55, 150, 56, 151],
X[56, 114, 57, 113], X[126, 48, 127, 47], X[125, 136, 126, 137], X[57, 48, 58, 49],
X[58, 136, 59, 135], X[124, 94, 125, 93], X[123, 34, 124, 35], X[59, 94, 60, 95],
X[60, 34, 61, 33], X[122, 8, 123, 7], X[121, 80, 122, 81], X[61, 8, 62, 9], X[62, 80, 63, 79],
X[147, 138, 148, 139], X[116, 138, 117, 137], X[148, 46, 149, 45], X[115, 46, 116, 47],
X[145, 36, 146, 37], X[118, 36, 119, 35], X[146, 92, 147, 91], X[117, 92, 118, 93],
X[143, 82, 144, 83], X[120, 82, 121, 81], X[144, 6, 145, 5], X[119, 6, 120, 7], X[43, 38, 44, 39],
X[140, 38, 141, 37], X[44, 90, 45, 89], X[139, 90, 140, 91], X[41, 84, 42, 85],
X[142, 84, 143, 83], X[42, 4, 43, 3], X[141, 4, 142, 5], X[87, 86, 88, 87], X[40, 86, 41, 85],
X[88, 2, 89, 1], X[39, 2, 40, 3], X[108, 14, 109, 13], X[107, 74, 108, 75], X[155, 14, 156, 15],
X[156, 74, 157, 73], X[30, 12, 31, 11], X[29, 76, 30, 77], X[97, 12, 98, 13], X[98, 76, 99, 75],
X[110, 52, 111, 51], X[109, 132, 110, 133], X[153, 52, 154, 53], X[154, 132, 155, 131],
X[101, 28, 102, 29], X[26, 28, 27, 27], X[102, 100, 103, 99], X[25, 100, 26, 101],
X[159, 106, 160, 107], X[104, 106, 105, 105], X[160, 158, 1, 157], X[103, 158, 104, 159]]

Out[•]= True

In[•]:=
```
DrawMorseLink@PD[X[15, 130, 16, 131], X[72, 130, 73, 129], X[16, 54, 17, 53],
  X[71, 54, 72, 55], X[70, 152, 71, 151], X[69, 112, 70, 113], X[17, 152, 18, 153],
  X[18, 112, 19, 111], X[68, 50, 69, 49], X[67, 134, 68, 135], X[19, 50, 20, 51],
  X[20, 134, 21, 133], X[66, 96, 67, 95], X[65, 32, 66, 33], X[21, 96, 22, 97],
  X[22, 32, 23, 31], X[64, 10, 65, 9], X[63, 78, 64, 79], X[23, 10, 24, 11],
  X[24, 78, 25, 77], X[128, 150, 129, 149], X[127, 114, 128, 115], X[55, 150, 56, 151],
  X[56, 114, 57, 113], X[126, 48, 127, 47], X[125, 136, 126, 137], X[57, 48, 58, 49],
  X[58, 136, 59, 135], X[124, 94, 125, 93], X[123, 34, 124, 35], X[59, 94, 60, 95],
  X[60, 34, 61, 33], X[122, 8, 123, 7], X[121, 80, 122, 81], X[61, 8, 62, 9],
  X[62, 80, 63, 79], X[147, 138, 148, 139], X[116, 138, 117, 137], X[148, 46, 149, 45],
  X[115, 46, 116, 47], X[145, 36, 146, 37], X[118, 36, 119, 35], X[146, 92, 147, 91],
  X[117, 92, 118, 93], X[143, 82, 144, 83], X[120, 82, 121, 81], X[144, 6, 145, 5],
  X[119, 6, 120, 7], X[43, 38, 44, 39], X[140, 38, 141, 37], X[44, 90, 45, 89],
  X[139, 90, 140, 91], X[41, 84, 42, 85], X[142, 84, 143, 83], X[42, 4, 43, 3],
  X[141, 4, 142, 5], X[87, 86, 88, 87], X[40, 86, 41, 85], X[88, 2, 89, 1], X[39, 2, 40, 3],
  X[108, 14, 109, 13], X[107, 74, 108, 75], X[155, 14, 156, 15], X[156, 74, 157, 73],
  X[30, 12, 31, 11], X[29, 76, 30, 77], X[97, 12, 98, 13], X[98, 76, 99, 75],
  X[110, 52, 111, 51], X[109, 132, 110, 133], X[153, 52, 154, 53], X[154, 132, 155, 131],
  X[101, 28, 102, 29], X[26, 28, 27, 27], X[102, 100, 103, 99], X[25, 100, 26, 101],
  X[159, 106, 160, 107], X[104, 106, 105, 105], X[160, 158, 1, 157], X[103, 158, 104, 159]]
```

Out[•]=



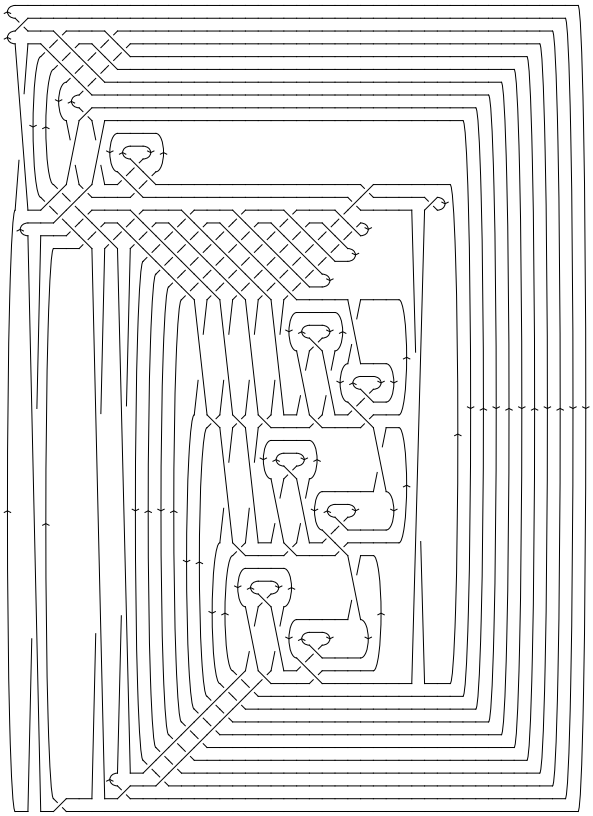
In[•]:=
```
TestKnotJones[Knot[7, 3]]
```

```
» PD[X[60, 18, 61, 17], X[59, 162, 60, 163], X[195, 18, 196, 19], X[196, 162, 197, 161],
  X[238, 16, 239, 15], X[237, 164, 238, 165], X[93, 16, 94, 17], X[94, 164, 95, 163],
  X[124, 14, 125, 13], X[123, 166, 124, 167], X[271, 14, 272, 15], X[272, 166, 273, 165],
  X[240, 194, 241, 193], X[239, 62, 240, 63], X[91, 194, 92, 195], X[92, 62, 93, 61],
  X[126, 192, 127, 191], X[125, 64, 126, 65], X[269, 192, 270, 193], X[270, 64, 271, 63],
  X[128, 90, 129, 89], X[127, 242, 128, 243], X[267, 90, 268, 91], X[268, 242, 269, 241],
  X[58, 96, 59, 95], X[57, 236, 58, 237], X[197, 96, 198, 97], X[198, 236, 199, 235],
  X[160, 98, 161, 97], X[159, 234, 160, 235], X[19, 98, 20, 99], X[20, 234, 21, 233],
  X[265, 232, 266, 233], X[130, 232, 131, 231], X[266, 100, 267, 99], X[129, 100, 130, 101],
  X[87, 230, 88, 231], X[244, 230, 245, 229], X[88, 102, 89, 101], X[243, 102, 244, 103],
  X[189, 228, 190, 229], X[66, 228, 67, 227], X[190, 104, 191, 103], X[65, 104, 66, 105],
  X[11, 226, 12, 227], X[168, 226, 169, 225], X[12, 106, 13, 105], X[167, 106, 168, 107],
  X[158, 200, 159, 199], X[157, 56, 158, 57], X[21, 200, 22, 201], X[22, 56, 23, 55],
  X[263, 54, 264, 55], X[132, 54, 133, 53], X[264, 202, 265, 201], X[131, 202, 132, 203],
  X[85, 52, 86, 53], X[246, 52, 247, 51], X[86, 204, 87, 203], X[245, 204, 246, 205],
  X[187, 50, 188, 51], X[68, 50, 69, 49], X[188, 206, 189, 205], X[67, 206, 68, 207], X[9, 48, 10, 49],
  X[170, 48, 171, 47], X[10, 208, 11, 207], X[169, 208, 170, 209], X[261, 156, 262, 157],
  X[134, 156, 135, 155], X[262, 24, 263, 23], X[133, 24, 134, 25], X[83, 154, 84, 155],
  X[248, 154, 249, 153], X[84, 26, 85, 25], X[247, 26, 248, 27], X[185, 152, 186, 153],
  X[70, 152, 71, 151], X[186, 28, 187, 27], X[69, 28, 70, 29], X[7, 150, 8, 151], X[172, 150, 173, 149],
  X[8, 30, 9, 29], X[171, 30, 172, 31], X[250, 260, 251, 259], X[249, 136, 250, 137],
  X[81, 260, 82, 261], X[82, 136, 83, 135], X[72, 258, 73, 257], X[71, 138, 72, 139],
  X[183, 258, 184, 259], X[184, 138, 185, 137], X[174, 256, 175, 255], X[173, 140, 174, 141],
  X[5, 256, 6, 257], X[6, 140, 7, 139], X[74, 80, 75, 79], X[73, 252, 74, 253], X[181, 80, 182, 81],
  X[182, 252, 183, 251], X[176, 78, 177, 77], X[175, 254, 176, 255], X[3, 78, 4, 79],
  X[4, 254, 5, 253], X[178, 180, 179, 179], X[177, 76, 178, 77], X[1, 180, 2, 181], X[2, 76, 3, 75],
  X[148, 34, 149, 33], X[147, 146, 148, 147], X[31, 34, 32, 35], X[32, 146, 33, 145],
  X[144, 38, 145, 37], X[143, 142, 144, 143], X[35, 38, 36, 39], X[36, 142, 37, 141],
  X[46, 212, 47, 211], X[45, 44, 46, 45], X[209, 212, 210, 213], X[210, 44, 211, 43],
  X[42, 216, 43, 215], X[41, 40, 42, 41], X[213, 216, 214, 217], X[214, 40, 215, 39],
  X[224, 110, 225, 109], X[223, 222, 224, 223], X[107, 110, 108, 111], X[108, 222, 109, 221],
  X[220, 114, 221, 113], X[219, 218, 220, 219], X[111, 114, 112, 115], X[112, 218, 113, 217],
  X[122, 276, 123, 275], X[121, 120, 122, 121], X[273, 276, 274, 277], X[274, 120, 275, 119],
  X[118, 280, 119, 279], X[117, 116, 118, 117], X[277, 280, 278, 1], X[278, 116, 279, 115]]
```

Out[•]=
```
True
```

```
In[•]:= DrawMorseLink@PD[X[60, 18, 61, 17], X[59, 162, 60, 163], X[195, 18, 196, 19],
  X[196, 162, 197, 161], X[238, 16, 239, 15], X[237, 164, 238, 165], X[93, 16, 94, 17],
  X[94, 164, 95, 163], X[124, 14, 125, 13], X[123, 166, 124, 167], X[271, 14, 272, 15],
  X[272, 166, 273, 165], X[240, 194, 241, 193], X[239, 62, 240, 63], X[91, 194, 92, 195],
  X[92, 62, 93, 61], X[126, 192, 127, 191], X[125, 64, 126, 65], X[269, 192, 270, 193],
  X[270, 64, 271, 63], X[128, 90, 129, 89], X[127, 242, 128, 243], X[267, 90, 268, 91],
  X[268, 242, 269, 241], X[58, 96, 59, 95], X[57, 236, 58, 237], X[197, 96, 198, 97],
  X[198, 236, 199, 235], X[160, 98, 161, 97], X[159, 234, 160, 235], X[19, 98, 20, 99],
  X[20, 234, 21, 233], X[265, 232, 266, 233], X[130, 232, 131, 231], X[266, 100, 267, 99],
  X[129, 100, 130, 101], X[87, 230, 88, 231], X[244, 230, 245, 229], X[88, 102, 89, 101],
  X[243, 102, 244, 103], X[189, 228, 190, 229], X[66, 228, 67, 227], X[190, 104, 191, 103],
  X[65, 104, 66, 105], X[11, 226, 12, 227], X[168, 226, 169, 225], X[12, 106, 13, 105],
  X[167, 106, 168, 107], X[158, 200, 159, 199], X[157, 56, 158, 57], X[21, 200, 22, 201],
  X[22, 56, 23, 55], X[263, 54, 264, 55], X[132, 54, 133, 53], X[264, 202, 265, 201],
  X[131, 202, 132, 203], X[85, 52, 86, 53], X[246, 52, 247, 51], X[86, 204, 87, 203],
  X[245, 204, 246, 205], X[187, 50, 188, 51], X[68, 50, 69, 49], X[188, 206, 189, 205],
  X[67, 206, 68, 207], X[9, 48, 10, 49], X[170, 48, 171, 47], X[10, 208, 11, 207],
  X[169, 208, 170, 209], X[261, 156, 262, 157], X[134, 156, 135, 155], X[262, 24, 263, 23],
  X[133, 24, 134, 25], X[83, 154, 84, 155], X[248, 154, 249, 153], X[84, 26, 85, 25],
  X[247, 26, 248, 27], X[185, 152, 186, 153], X[70, 152, 71, 151], X[186, 28, 187, 27],
  X[69, 28, 70, 29], X[7, 150, 8, 151], X[172, 150, 173, 149], X[8, 30, 9, 29],
  X[171, 30, 172, 31], X[250, 260, 251, 259], X[249, 136, 250, 137], X[81, 260, 82, 261],
  X[82, 136, 83, 135], X[72, 258, 73, 257], X[71, 138, 72, 139], X[183, 258, 184, 259],
  X[184, 138, 185, 137], X[174, 256, 175, 255], X[173, 140, 174, 141], X[5, 256, 6, 257],
  X[6, 140, 7, 139], X[74, 80, 75, 79], X[73, 252, 74, 253], X[181, 80, 182, 81],
  X[182, 252, 183, 251], X[176, 78, 177, 77], X[175, 254, 176, 255], X[3, 78, 4, 79],
  X[4, 254, 5, 253], X[178, 180, 179, 179], X[177, 76, 178, 77], X[1, 180, 2, 181],
  X[2, 76, 3, 75], X[148, 34, 149, 33], X[147, 146, 148, 147], X[31, 34, 32, 35],
  X[32, 146, 33, 145], X[144, 38, 145, 37], X[143, 142, 144, 143], X[35, 38, 36, 39],
  X[36, 142, 37, 141], X[46, 212, 47, 211], X[45, 44, 46, 45], X[209, 212, 210, 213],
  X[210, 44, 211, 43], X[42, 216, 43, 215], X[41, 40, 42, 41], X[213, 216, 214, 217],
  X[214, 40, 215, 39], X[224, 110, 225, 109], X[223, 222, 224, 223], X[107, 110, 108, 111],
  X[108, 222, 109, 221], X[220, 114, 221, 113], X[219, 218, 220, 219],
  X[111, 114, 112, 115], X[112, 218, 113, 217], X[122, 276, 123, 275],
  X[121, 120, 122, 121], X[273, 276, 274, 277], X[274, 120, 275, 119],
  X[118, 280, 119, 279], X[117, 116, 118, 117], X[277, 280, 278, 1], X[278, 116, 279, 115]]
```

Out[•]=



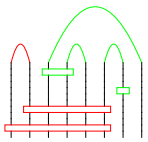
In[•]:= `TestKnotJones /@ AllKnots[{3, 7}]`

Out[•]= $Aborted

In[•]:= `Knot2Bandtangle[PD@Knot[8, 20]]`
`DrawTangle@Knot2Bandtangle[PD@Knot[8, 20]]`

Out[•]= $\{\{G_1[3, 4], G_1[7], G_{-1}[2, 3, 4, 5, 6], G_{-1}[1, 2, 3, 4, 5, 6]\}, \{1_{-1}, 1_{-1}, 2_1, 3_1, 3_1, 4_1, 4_1, 2_1\}\}$

Out[•]=



In[•]:= `TestKnotAlex[Knot[8, 20]]`

Out[•]= True

In[ ]:= **TestKnotAlex /@ AllKnots[{10, 10}]**

Out[ ]= {True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True,
$-31+\frac{2}{t^3}-\frac{10}{t^2}+\frac{24}{t}+24\,t-10\,t^2+2\,t^3 == -25+\frac{2}{t^3}-\frac{9}{t^2}+\frac{20}{t}+20\,t-9\,t^2+2\,t^3$, True, True,
$29-\frac{2}{t^3}+\frac{10}{t^2}-\frac{22}{t}-22\,t+10\,t^2-2\,t^3 == 23-\frac{2}{t^3}+\frac{9}{t^2}-\frac{18}{t}-18\,t+9\,t^2-2\,t^3$, True, True,
True, True, $27-\frac{3}{t^3}+\frac{13}{t^2}-\frac{23}{t}-23\,t+13\,t^2-3\,t^3 == 25-\frac{2}{t^3}+\frac{10}{t^2}-\frac{20}{t}-20\,t+10\,t^2-2\,t^3$,
$-21+\frac{2}{t^3}-\frac{8}{t^2}+\frac{17}{t}+17\,t-8\,t^2+2\,t^3 == -17+\frac{2}{t^3}-\frac{8}{t^2}+\frac{15}{t}+15\,t-8\,t^2+2\,t^3$, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, $-41+\frac{3}{t^3}-\frac{15}{t^2}+\frac{33}{t}+33\,t-15\,t^2+3\,t^3 == -33+\frac{2}{t^3}-\frac{11}{t^2}+\frac{26}{t}+26\,t-11\,t^2+2\,t^3$,
True, True, True, True, True, True, True,
$-41+\frac{2}{t^3}-\frac{12}{t^2}+\frac{31}{t}+31\,t-12\,t^2+2\,t^3 == -35+\frac{2}{t^3}-\frac{11}{t^2}+\frac{27}{t}+27\,t-11\,t^2+2\,t^3$, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True, True,
True, True, True, True, True, True, True, $3-\frac{1}{t}-t == -9-\frac{2}{t^2}+\frac{7}{t}+7\,t-2\,t^2$, True, True, True,
True, True, True, True, True, True, True, True, True, True, True, True, True, True, True}