

Pensieve header: Dalvit's enumeration of w -knotted spheres with up to 4 crossings. Output from Jeremy Green's C program in Mathematica format.

Draw diagrams of w -knotted arcs from codes obtained modifying Green's virtual knots enumerator.

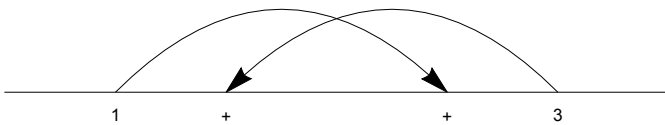
```
(*SetDirectory["C:\\drorbn\\AcademicPensieve\\2015-03" ]*)

A_List \ B_List := Complement[A, B];

Draw[w_wLDiag | w_wCDiag] := Module[{n, w1},
  n = Length[w];
  w1 = Abs /@ w;
  Graphics[{
    Line[{{0, 0}, {n + 1, 0}}],
    Table[
      {
        Arrow[BezierCurve[
          {{w1[[j]] - 0.5, 0}, {(w1[[j]] + j - 0.5) / 2, 0.5 Abs[j - w1[[j]] + 0.5]}, {j, 0}}]],
        Text[If[w[[j]] > 0, "+", "-"], {j, -0.1}],
        Text[w1[[j]], {w1[[j]] - 0.5, -0.1}]
      },
      {j, n}
    ]
  ]];

Draw[expr_] := expr /. w_wLDiag | w_wCDiag -> Draw[w]

Draw[wLDiag[3, 1]]
```



```
AllLinearDiagrams[n_] := Flatten@Table[
  wLDiag@@@Tuples[Range[k + 1] ∪ (-Range[k + 1]), k],
  {k, 0, n}
]
```

AllLinearDiagrams[2]

```
{wLDiag[], wLDiag[-2], wLDiag[-1], wLDiag[1], wLDiag[2], wLDiag[-3, -3],
  wLDiag[-3, -2], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[-3, 2], wLDiag[-3, 3],
  wLDiag[-2, -3], wLDiag[-2, -2], wLDiag[-2, -1], wLDiag[-2, 1], wLDiag[-2, 2],
  wLDiag[-2, 3], wLDiag[-1, -3], wLDiag[-1, -2], wLDiag[-1, -1], wLDiag[-1, 1],
  wLDiag[-1, 2], wLDiag[-1, 3], wLDiag[1, -3], wLDiag[1, -2], wLDiag[1, -1],
  wLDiag[1, 1], wLDiag[1, 2], wLDiag[1, 3], wLDiag[2, -3], wLDiag[2, -2],
  wLDiag[2, -1], wLDiag[2, 1], wLDiag[2, 2], wLDiag[2, 3], wLDiag[3, -3],
  wLDiag[3, -2], wLDiag[3, -1], wLDiag[3, 1], wLDiag[3, 2], wLDiag[3, 3]}
```

```
RemoveR1[w_wLDiag] := Module[{j, k = 0},
  Do[If[MemberQ[{j, j + 1}, Abs[w[[j]]], k = j], {j, Length[w]}];
  If[k == 0, w,
    Delete[w, k] /. j_Integer /; Abs[j] > k => Sign[j] (Abs[j] - 1)
  ]
]
```

```
RemoveR1[wLDiag[-4, 1, 3, -4]]
```

```
wLDiag[-4, 1, 3]
```

RemoveR1 /@AllLinearDiagrams[2]

```
{wLDiag[], wLDiag[], wLDiag[], wLDiag[], wLDiag[], wLDiag[-2], wLDiag[-2],
  wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[-2], wLDiag[-2], wLDiag[-2], wLDiag[-2],
  wLDiag[-1], wLDiag[1], wLDiag[-2], wLDiag[-2], wLDiag[-1], wLDiag[-1], wLDiag[-1],
  wLDiag[1], wLDiag[-1], wLDiag[-1], wLDiag[1], wLDiag[1], wLDiag[-1], wLDiag[1],
  wLDiag[1], wLDiag[1], wLDiag[2], wLDiag[2], wLDiag[-1], wLDiag[1], wLDiag[2],
  wLDiag[2], wLDiag[2], wLDiag[2], wLDiag[3, -1], wLDiag[3, 1], wLDiag[2], wLDiag[2]}
```

```
RemoveR1s[w_wLDiag | w_wCDiag] := FixedPoint[RemoveR1, w]
```

RemoveR1s /@AllLinearDiagrams[2] // Union

```
{wLDiag[], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[3, -1], wLDiag[3, 1]}
```

```
RemoveR2[w_wLDiag] := Module[{j, k = 0},
```

```
  Do[
    If[w[[j]] + w[[j + 1]] == 0 && !MemberQ[Abs[List@@w], j + 1], k = j], {j, Length[w] - 1};
  If[k == 0, w,
    Delete[w, {{k}, {k + 1}}] /. j_Integer /; Abs[j] > k => Sign[j] (Abs[j] - 2)
  ]
]
```

```
wLDiag[2, -2] // RemoveR2
```

```
wLDiag[2, -2]
```

RemoveR2 /@ **AllLinearDiagrams**[2]

```
{wLDiag[], wLDiag[-2], wLDiag[-1], wLDiag[1], wLDiag[2], wLDiag[-3, -3],
  wLDiag[-3, -2], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[-3, 2], wLDiag[],
  wLDiag[-2, -3], wLDiag[-2, -2], wLDiag[-2, -1], wLDiag[-2, 1], wLDiag[-2, 2],
  wLDiag[-2, 3], wLDiag[-1, -3], wLDiag[-1, -2], wLDiag[-1, -1], wLDiag[],
  wLDiag[-1, 2], wLDiag[-1, 3], wLDiag[1, -3], wLDiag[1, -2], wLDiag[],
  wLDiag[1, 1], wLDiag[1, 2], wLDiag[1, 3], wLDiag[2, -3], wLDiag[2, -2],
  wLDiag[2, -1], wLDiag[2, 1], wLDiag[2, 2], wLDiag[2, 3], wLDiag[],
  wLDiag[3, -2], wLDiag[3, -1], wLDiag[3, 1], wLDiag[3, 2], wLDiag[3, 3]}
```

AllLinearDiagrams[2]

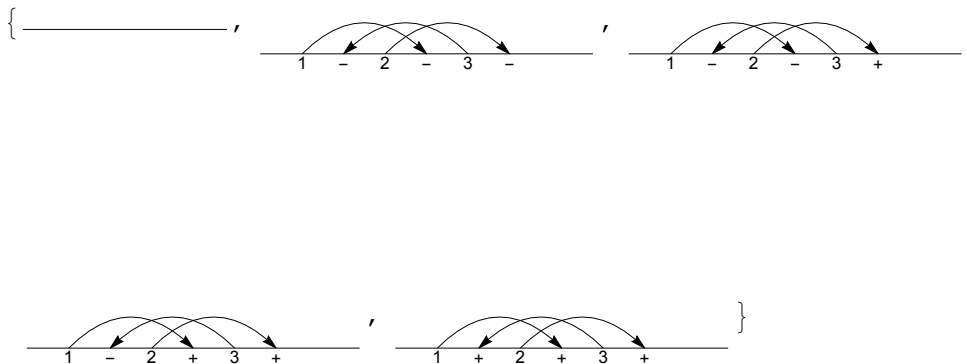
```
{wLDiag[], wLDiag[-2], wLDiag[-1], wLDiag[1], wLDiag[2], wLDiag[-3, -3],
  wLDiag[-3, -2], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[-3, 2], wLDiag[-3, 3],
  wLDiag[-2, -3], wLDiag[-2, -2], wLDiag[-2, -1], wLDiag[-2, 1], wLDiag[-2, 2],
  wLDiag[-2, 3], wLDiag[-1, -3], wLDiag[-1, -2], wLDiag[-1, -1], wLDiag[-1, 1],
  wLDiag[-1, 2], wLDiag[-1, 3], wLDiag[1, -3], wLDiag[1, -2], wLDiag[1, -1],
  wLDiag[1, 1], wLDiag[1, 2], wLDiag[1, 3], wLDiag[2, -3], wLDiag[2, -2],
  wLDiag[2, -1], wLDiag[2, 1], wLDiag[2, 2], wLDiag[2, 3], wLDiag[3, -3],
  wLDiag[3, -2], wLDiag[3, -1], wLDiag[3, 1], wLDiag[3, 2], wLDiag[3, 3]}
```

Select[**AllLinearDiagrams**[2], (**#** != **RemoveR2**[**#**]) &]

```
{wLDiag[-3, 3], wLDiag[-1, 1], wLDiag[1, -1], wLDiag[3, -3]}
```

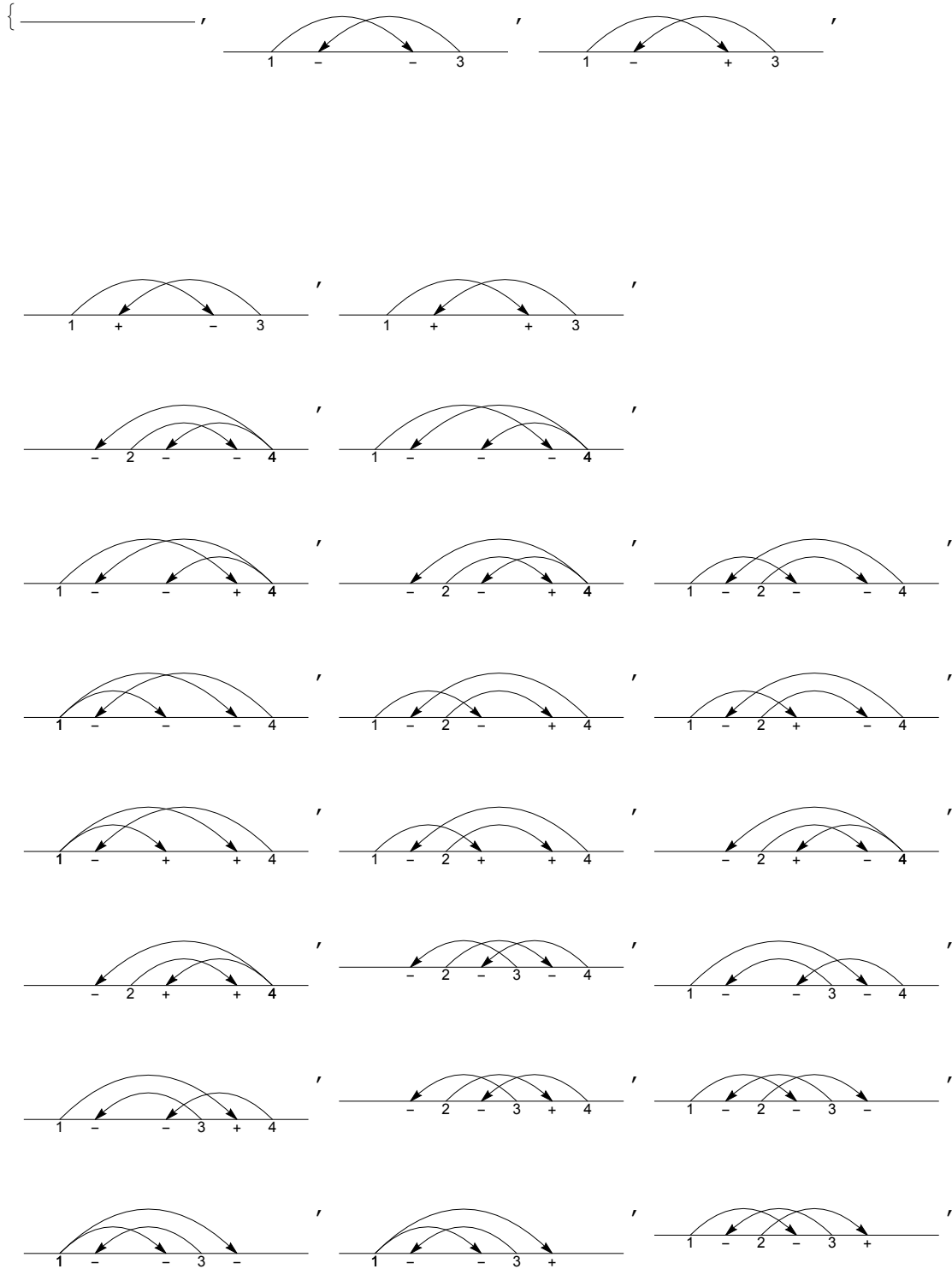
RemoveR12s[**w_wLDiag** | **w_wCDiag**] := **FixedPoint**[**RemoveR2**[**RemoveR1**[**#**]] &, **w**]

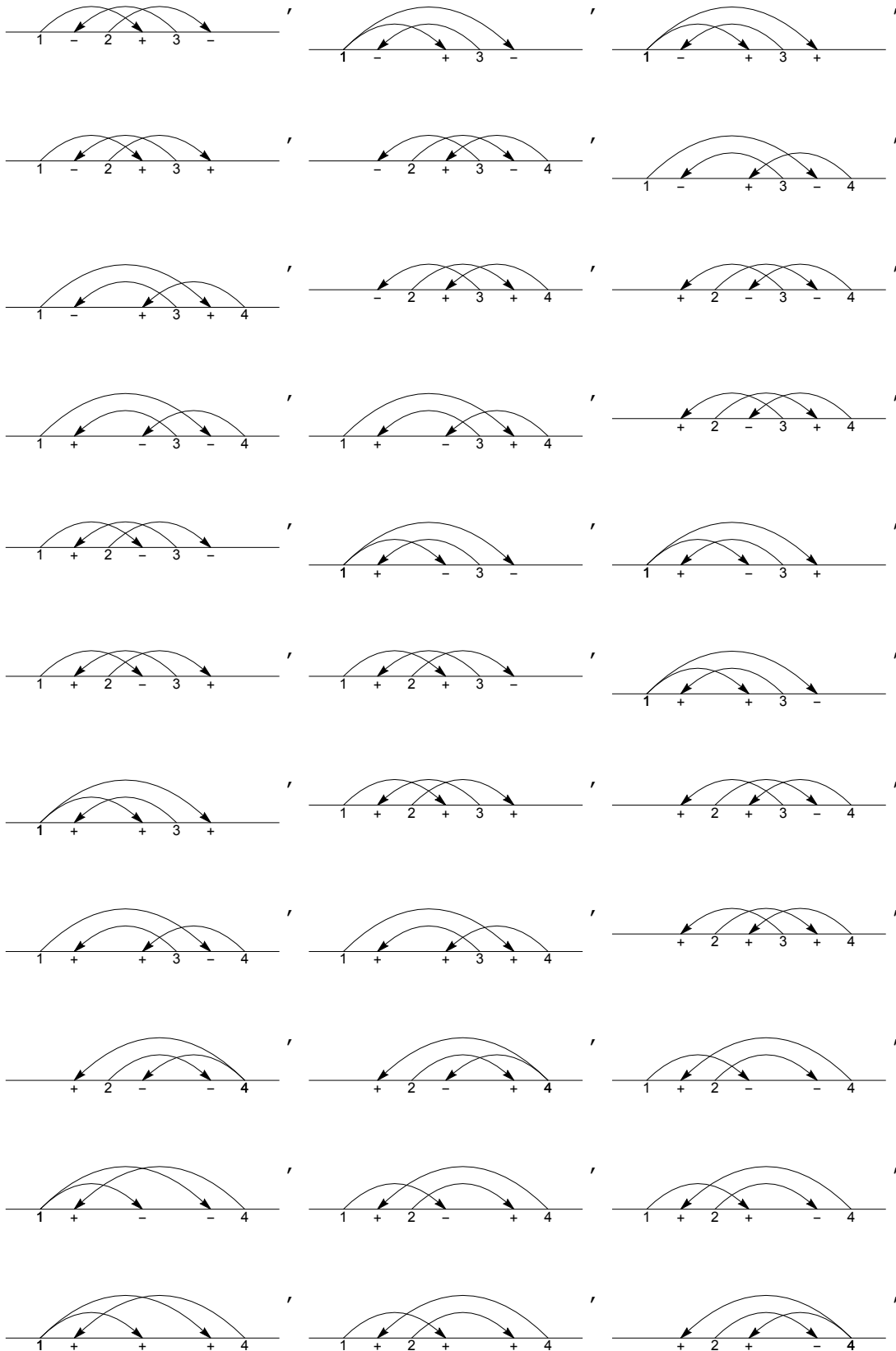
Union[**RemoveR12s** /@ **AllLinearDiagrams**[3]] // **Draw**

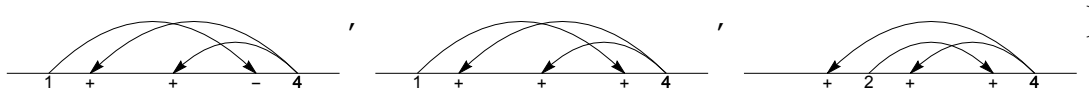


RF[**w_wLDiag**] := **RemoveR12s**[**w**];

Union[**RF** /@ **AllLinearDiagrams**[3]] // **Draw**

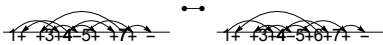






```
wLDiag /:
Resolve[wLDiag[R3[top_, mid_, bot_, s1_, s2_, s3_], ts_]] := UndirectedEdge[
  RF@ReplacePart[wLDiag@ts, {bot + (1 - s3) / 2 -> s2 s3 top,
    bot + (1 + s3) / 2 -> s1 s3 (mid + 1), mid -> s2 top}],
  RF@ReplacePart[wLDiag@ts, {bot + (1 - s3) / 2 -> s1 s3 mid,
    bot + (1 + s3) / 2 -> s2 s3 top, mid -> s2 top}]
];
```

```
Resolve@wLDiag[R3[4, 6, 1, 1, 1, 1], 0, 0, +1, -3, +4, 0, +5, -7] // Draw
```



```
AllLinearR3s[n_] /; n < 3 := {};
```

```
AllLinearR3s[n_] := Flatten@Table[
  Prepend[
    ReplacePart[wLDiag@@Table[0, {n}],
      Thread[Range[n] \ {bot, bot + 1, mid} -> #]],
    R3[top, mid, bot, s1, s2, s3]
  ] & /@ Tuples[Range[-n - 1, n + 1] \ {-bot - 1, 0, bot + 1}, n - 3],
  {bot, Range[n - 1]},
  {mid, Range[n] \ {bot, bot + 1}}, {top, Range[n + 1] \ {bot + 1}},
  {s1, {-1, 1}}, {s2, {-1, 1}}, {s3, {-1, 1}}
];
```

```
Union[RF /@ AllLinearDiagrams[4]]
```

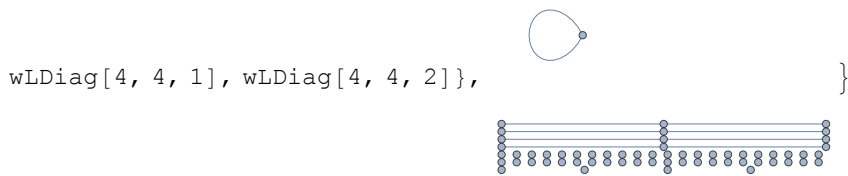
```
{wLDiag[], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[3, -1], wLDiag[3, 1],
wLDiag[-4, -4, -2], wLDiag[-4, -4, -1], ... 1163 ..., wLDiag[5, 5, 2, 3],
wLDiag[5, 5, 5, -3], wLDiag[5, 5, 5, -2], wLDiag[5, 5, 5, -1],
wLDiag[5, 5, 5, 1], wLDiag[5, 5, 5, 2], wLDiag[5, 5, 5, 3]}
```

large output	show less	show more	show all	set size limit...
--------------	---------------------------	---------------------------	--------------------------	-----------------------------------

```
wLinearK[n_] :=
  wLinearK[n] = Module[{VS, vs, rule, ES, es, $k}, VS = AllLinearDiagrams[n];
    $k = 0; vs = Union[(++$k; RF[#]) & /@VS];
    Print[Length@vs];
    rule = Dispatch[Thread[vs → Range[Length@vs]]];
    ES = AllLinearR3s[n];
    Print[Length@ES];
    $k = 0; es = Union[(++$k; Resolve[#] /. rule) & /@ES];
    {vs, Graph[Range[Length@vs], es]};
wLinearK[n_, k_] := wLinearK[n, k] = Module[{vs, g, cc}, {vs, g} = wLinearK[n];
  cc = ConnectedComponents[g];
  Select[Table[First@MinimalBy[vs[[#]] & /@c, Length], {c, cc}], Length[#] == k &]
```

wLinearK[3]

```
{wLDiag[], wLDiag[-3, -1], wLDiag[-3, 1], wLDiag[3, -1],
  wLDiag[3, 1], wLDiag[-4, -4, -2], wLDiag[-4, -4, -1], wLDiag[-4, -4, 1],
  wLDiag[-4, -4, 2], wLDiag[-4, -1, -2], wLDiag[-4, -1, -1],
  wLDiag[-4, -1, 2], wLDiag[-4, 1, -2], wLDiag[-4, 1, 1], wLDiag[-4, 1, 2],
  wLDiag[-4, 4, -2], wLDiag[-4, 4, 2], wLDiag[-3, -4, -2], wLDiag[-3, -4, -1],
  wLDiag[-3, -4, 1], wLDiag[-3, -4, 2], wLDiag[-3, -1, -2], wLDiag[-3, -1, -1],
  wLDiag[-3, -1, 1], wLDiag[-3, -1, 2], wLDiag[-3, 1, -2], wLDiag[-3, 1, -1],
  wLDiag[-3, 1, 1], wLDiag[-3, 1, 2], wLDiag[-3, 4, -2], wLDiag[-3, 4, -1],
  wLDiag[-3, 4, 1], wLDiag[-3, 4, 2], wLDiag[3, -4, -2], wLDiag[3, -4, -1],
  wLDiag[3, -4, 1], wLDiag[3, -4, 2], wLDiag[3, -1, -2], wLDiag[3, -1, -1],
  wLDiag[3, -1, 1], wLDiag[3, -1, 2], wLDiag[3, 1, -2], wLDiag[3, 1, -1],
  wLDiag[3, 1, 1], wLDiag[3, 1, 2], wLDiag[3, 4, -2], wLDiag[3, 4, -1],
  wLDiag[3, 4, 1], wLDiag[3, 4, 2], wLDiag[4, -4, -2], wLDiag[4, -4, 2],
  wLDiag[4, -1, -2], wLDiag[4, -1, -1], wLDiag[4, -1, 2], wLDiag[4, 1, -2],
  wLDiag[4, 1, 1], wLDiag[4, 1, 2], wLDiag[4, 4, -2], wLDiag[4, 4, -1],
```



```
HMirror[w_wCDiag | w_wLDiag] := RF[(-#) & /@ w]
```

```
HMirror[wLDiag[-3, 4, -1, -2]] // RF
```

```
wLDiag[3, -4, 1, 2]
```

```
Position[wLinearK[3], wLDiag[3, 1]]
```

```
{{1, 5}}
```

```

ReduceMod[w_, {vs_, g_}] := Module[{p, c, cc},
  {{p}} = Position[vs, w];
  c = Position[cc = ConnectedComponents[g], p][[1, 1]];
  First@MinimalBy[vs[[#]] & /@ cc[[c]], Length]
]

ReduceMod[{wLDiag[3, 1, -2] // RF}, wLinearK[3]]
wLDiag[3, 1, -2]

hms = ReduceMod[HMirror[#] // RF, wLinearK[3]] & /@ wLinearK[3, 2]
{wLDiag[-3, 1], wLDiag[3, -1], wLDiag[-3, -1], wLDiag[3, 1]}

(Position[wLinearK[3, 2], #][[1, 1]]) & /@ hms
{2, 1, 4, 3}

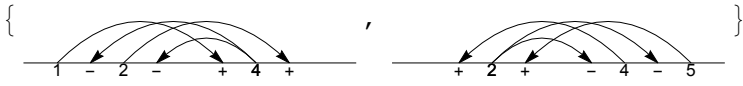
PermutationCycles[
  {1, 3, 2, 4, 8, 7, 6, 5, 13, 11, 10, 14, 9, 12, 19, 18, 20, 16, 15, 17, 21, 25, 24, 23, 22}]
Cycles[{{2, 3}, {5, 8}, {6, 7}, {9, 13}, {10, 11},
  {12, 14}, {15, 19}, {16, 18}, {17, 20}, {22, 25}, {23, 24}}]

Cycles[{{2, 3}, {5, 8}, {6, 7}, {9, 13}, {10, 11}, {12, 14},
  {15, 19}, {16, 18}, {17, 20}, {22, 25}, {23, 24}}] // First // Length
11

wLDiag /: Reverse[w_wLDiag] := Module[{n2},
  n2 = Length[w] + 2;
  wLDiag@@Reverse[(Sign[#] * (n2 - Abs[#]))] & /@ List@@w]
];

Reverse[wLDiag[-3, 1, -1]]
wLDiag[-4, 4, -2]

{wLDiag[-4, -4, 1, 2], Reverse[wLDiag[-4, -4, 1, 2]]} // Draw

{

}

rvs = ReduceMod[Reverse[#] // RF, wLinearK[3]] & /@ wLinearK[3, 2]
{wLDiag[-3, 1], wLDiag[3, -1], wLDiag[3, 1], wLDiag[-3, -1]}

PermutationCycles[(Position[wLinearK[3, 2], #][[1, 1]]) & /@ rvs]
Cycles[{{1, 2}}]

Cycles[{{2, 3}, {5, 8}, {6, 7}, {9, 13}, {10, 11},
  {12, 14}, {15, 19}, {16, 18}, {17, 20}, {22, 25}, {23, 24}}]

```



```
{1}, {2, 3}, {4}, {5, 6, 7, 8}, {9, 13}, {10, 11, 12, 14},
{15, 17, 19, 20}, {16, 18}, {21}, {22, 23, 24, 25} // Length
```

```
10
```

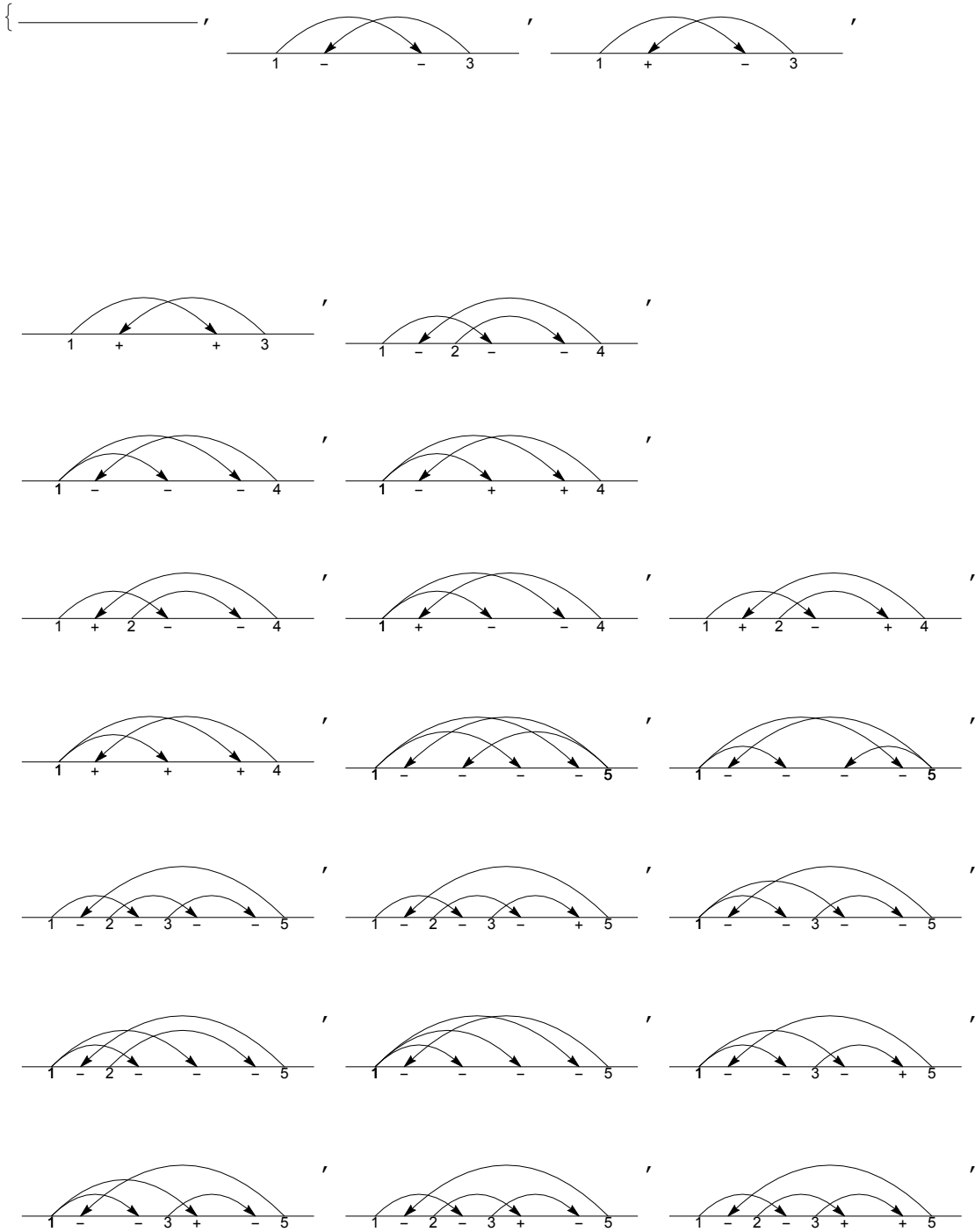
```
AllCapped74 :=
```

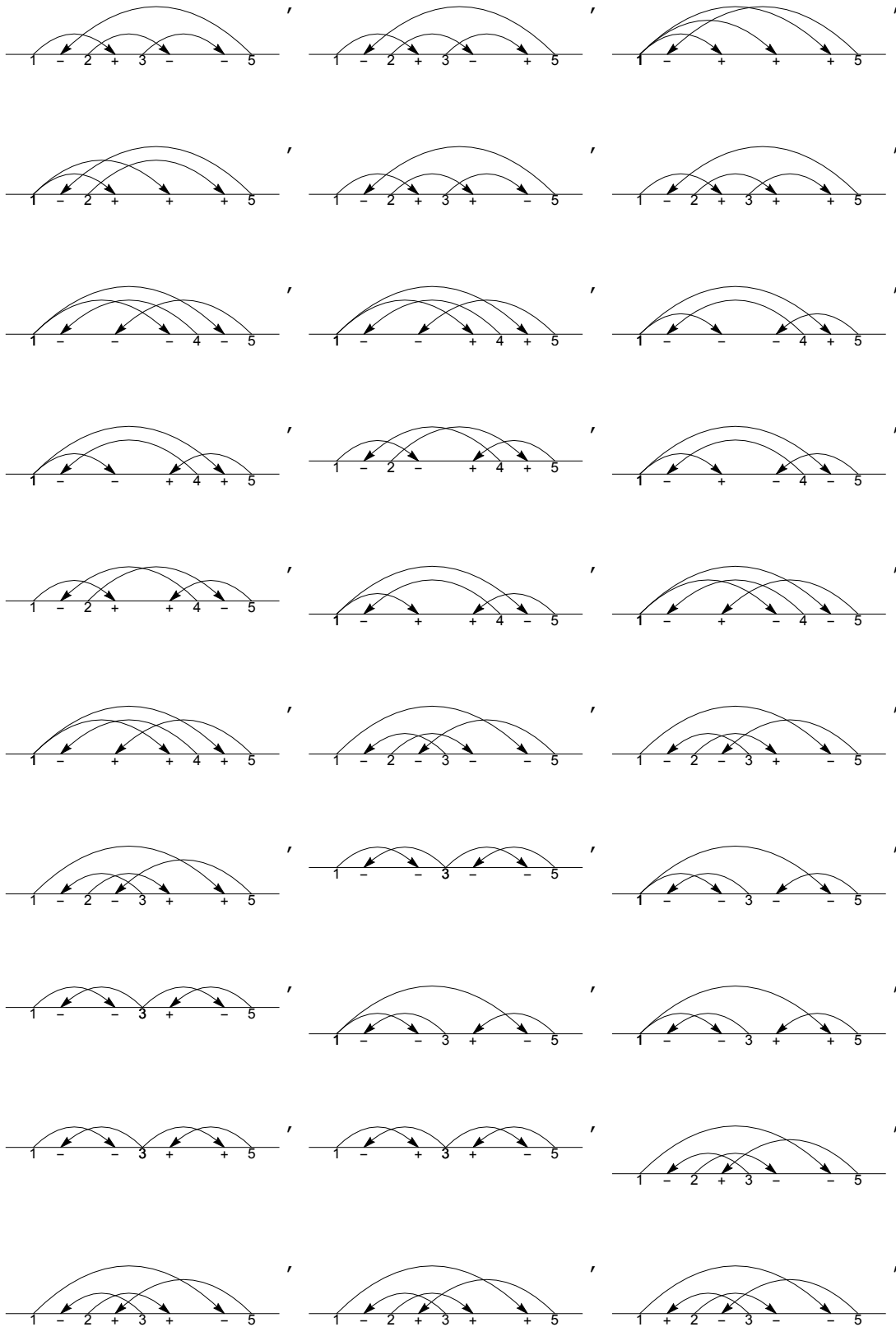
```
Union[{ wLDiag[], wLDiag[-3, -1], wLDiag[3, -1], wLDiag[3, 1], wLDiag[-4, -1, -1],
wLDiag[4, -1, -1], wLDiag[-4, 1, 1], wLDiag[4, 1, 1], wLDiag[-4, -1, -2],
wLDiag[4, -1, -2], wLDiag[4, -1, 2], wLDiag[-5, -1, -1, -1],
wLDiag[5, -1, -1, -1], wLDiag[-5, 1, 1, 1], wLDiag[5, 1, 1, 1],
wLDiag[-3, -1, -5, -1], wLDiag[-3, -1, 5, -1], wLDiag[3, -1, -5, -1],
wLDiag[3, -1, 5, -1], wLDiag[-3, -1, 5, 1], wLDiag[3, -1, -5, 1],
wLDiag[3, -1, 5, 1], wLDiag[3, 1, -5, -1], wLDiag[3, 1, 5, -1], wLDiag[3, 1, 5, 1],
wLDiag[-5, -1, -1, -3], wLDiag[-5, -1, -1, 3], wLDiag[5, -1, -1, -3],
wLDiag[5, -1, -1, 3], wLDiag[-5, -1, 1, -3], wLDiag[5, -1, 1, -3],
wLDiag[5, 1, 1, -3], wLDiag[5, 1, 1, 3], wLDiag[-5, -1, -5, -1],
wLDiag[5, -1, 5, -1], wLDiag[5, 1, 5, 1], wLDiag[-4, -1, -5, 1],
wLDiag[-4, -1, 5, 1], wLDiag[4, -1, -5, 1], wLDiag[4, -1, 5, 1],
wLDiag[-4, 1, -5, -1], wLDiag[-4, 1, 5, -1], wLDiag[4, 1, -5, -1],
wLDiag[4, 1, 5, -1], wLDiag[-5, -1, -1, -2], wLDiag[5, -1, -1, -2],
wLDiag[-5, 1, 1, 2], wLDiag[5, 1, 1, 2], wLDiag[-5, -5, -1, -1],
wLDiag[5, 5, -1, -1], wLDiag[5, 5, 1, 1], wLDiag[-4, -5, -1, -1],
wLDiag[-4, 5, -1, -1], wLDiag[4, -5, -1, -1], wLDiag[4, 5, -1, -1],
wLDiag[-4, -5, 1, 1], wLDiag[-4, 5, 1, 1], wLDiag[4, 5, 1, 1],
wLDiag[-3, -1, -5, -3], wLDiag[-3, -1, 5, -3], wLDiag[-3, -1, 5, 3],
wLDiag[3, -1, 5, -3], wLDiag[3, -1, -5, 3], wLDiag[3, -1, 5, 3],
wLDiag[-3, 1, 5, -3], wLDiag[3, 1, 5, 3], wLDiag[-5, -1, -2, -3],
wLDiag[-5, -1, -2, 3], wLDiag[-5, -1, 2, -3], wLDiag[-5, -1, 2, 3],
wLDiag[5, -1, -2, -3], wLDiag[5, -1, -2, 3], wLDiag[5, -1, 2, -3],
wLDiag[5, -1, 2, 3], wLDiag[-5, 1, -2, -3], wLDiag[-5, 1, -2, 3],
wLDiag[-5, 1, 2, -3], wLDiag[-5, 1, 2, 3], wLDiag[5, 1, -2, -3],
wLDiag[5, 1, -2, 3], wLDiag[5, 1, 2, -3], wLDiag[5, 1, 2, 3],
wLDiag[-4, -1, 5, 2], wLDiag[4, -1, -5, 2], wLDiag[-4, 1, 5, -2],
wLDiag[5, -1, -2, -2], wLDiag[-3, -5, -2, -1], wLDiag[-3, 5, -2, -1],
wLDiag[-3, -5, 2, -1], wLDiag[-3, 5, 2, -1], wLDiag[3, -5, -2, -1],
wLDiag[3, 5, -2, -1], wLDiag[3, 5, 2, -1], wLDiag[-3, -5, 2, 1],
wLDiag[-3, 5, 2, 1], wLDiag[3, -5, 2, 1], wLDiag[3, 5, 2, 1]};
```

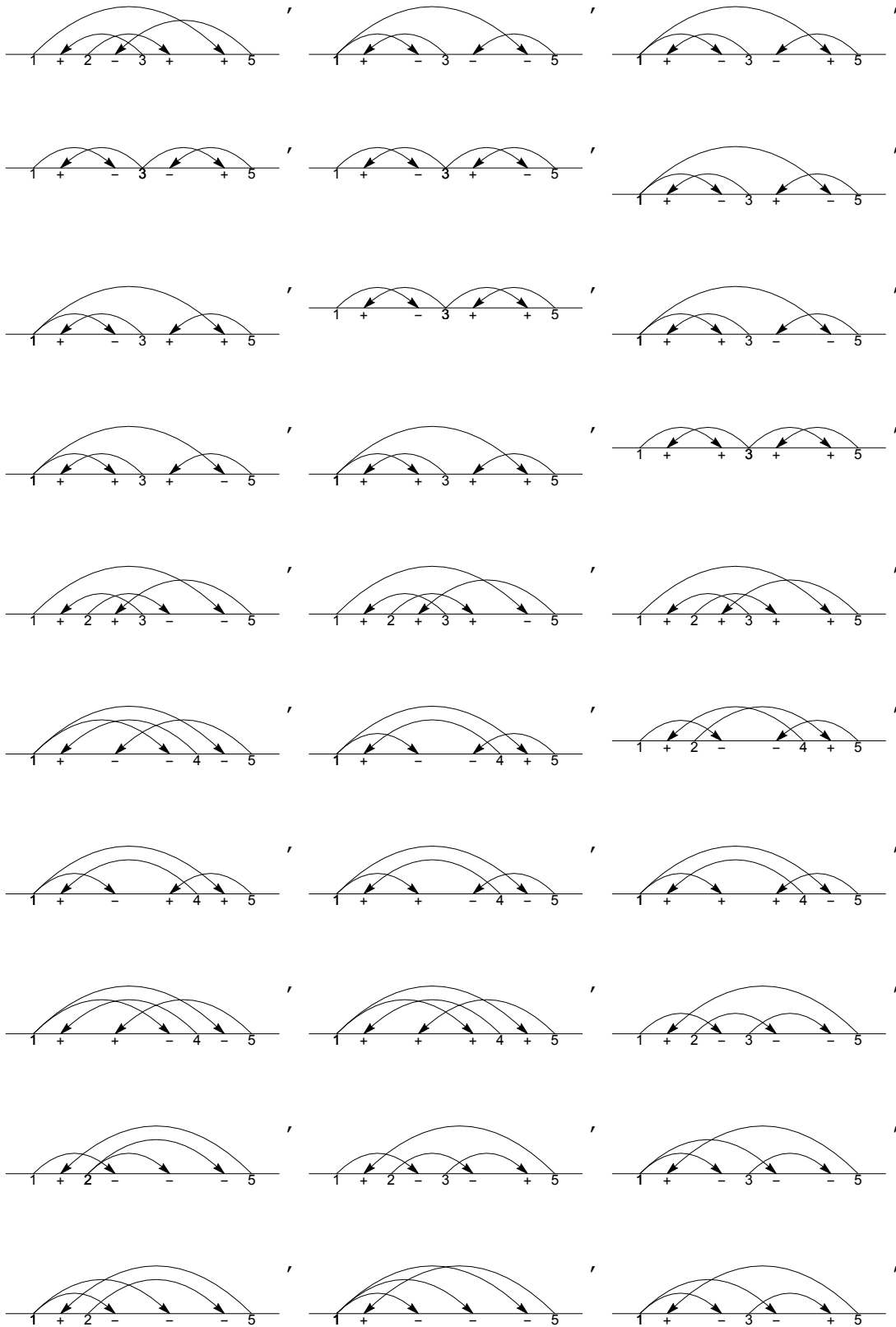
```
Length[AllCapped74]
```

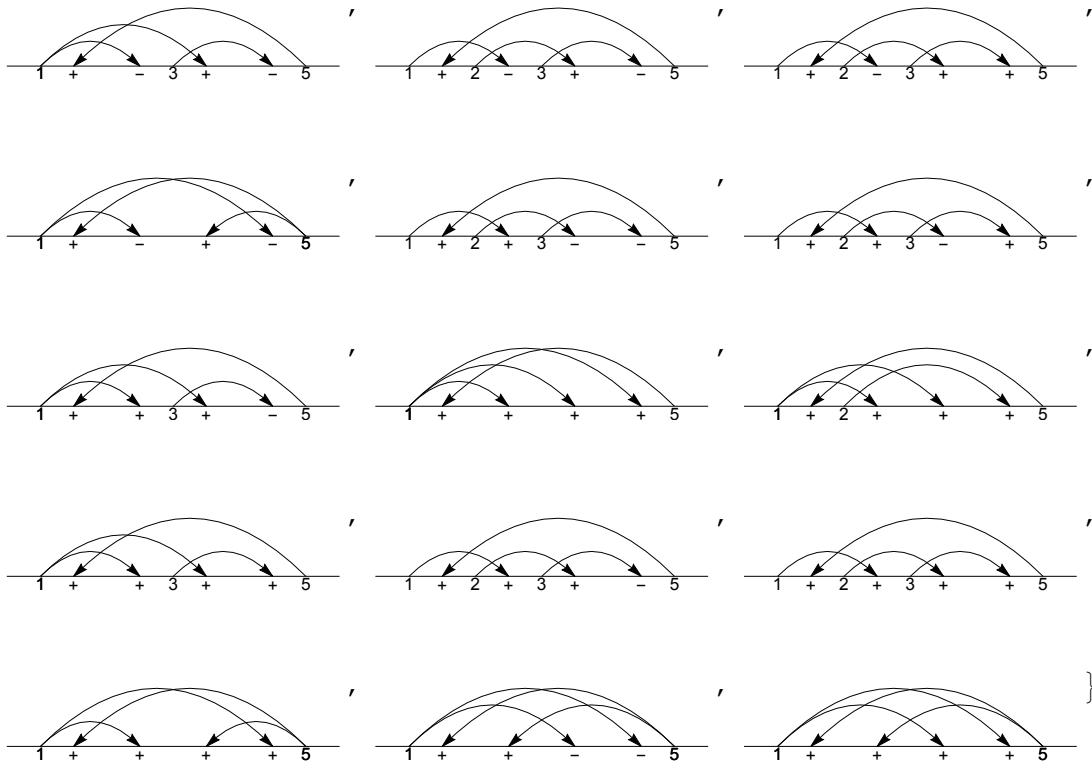
```
97
```

```
Draw[AllCapped74]
```









AllCapped74RF = Union[RF /@ AllCapped74]

```
{wLDiag[], wLDiag[-3, -1], wLDiag[3, -1], wLDiag[3, 1], wLDiag[-4, -1, -2],
wLDiag[-4, -1, -1], wLDiag[-4, 1, 1], wLDiag[4, -1, -2], wLDiag[4, -1, -1],
wLDiag[4, -1, 2], wLDiag[4, 1, 1], wLDiag[-5, -5, -1, -1], wLDiag[-5, -1, -5, -1],
wLDiag[-5, -1, -2, -3], wLDiag[-5, -1, -2, 3], wLDiag[-5, -1, -1, -3],
wLDiag[-5, -1, -1, -2], wLDiag[-5, -1, -1, -1], wLDiag[-5, -1, -1, 3],
wLDiag[-5, -1, 1, -3], wLDiag[-5, -1, 2, -3], wLDiag[-5, -1, 2, 3],
wLDiag[-5, 1, -2, -3], wLDiag[-5, 1, -2, 3], wLDiag[-5, 1, 1, 1], wLDiag[-5, 1, 1, 2],
wLDiag[-5, 1, 2, -3], wLDiag[-5, 1, 2, 3], wLDiag[-4, -5, -1, -1],
wLDiag[-4, -5, 1, 1], wLDiag[-4, -1, -5, 1], wLDiag[-4, -1, 5, 1],
wLDiag[-4, -1, 5, 2], wLDiag[-4, 1, -5, -1], wLDiag[-4, 1, 5, -2],
wLDiag[-4, 1, 5, -1], wLDiag[-4, 5, -1, -1], wLDiag[-4, 5, 1, 1],
wLDiag[-3, -5, -2, -1], wLDiag[-3, -5, 2, -1], wLDiag[-3, -5, 2, 1],
wLDiag[-3, -1, -5, -3], wLDiag[-3, -1, -5, -1], wLDiag[-3, -1, 5, -3],
wLDiag[-3, -1, 5, -1], wLDiag[-3, -1, 5, 1], wLDiag[-3, -1, 5, 3],
wLDiag[-3, 1, 5, -3], wLDiag[-3, 5, -2, -1], wLDiag[-3, 5, 2, -1],
wLDiag[-3, 5, 2, 1], wLDiag[3, -5, -2, -1], wLDiag[3, -5, 2, 1],
wLDiag[3, -1, -5, -1], wLDiag[3, -1, -5, 1], wLDiag[3, -1, -5, 3],
wLDiag[3, -1, 5, -3], wLDiag[3, -1, 5, -1], wLDiag[3, -1, 5, 1], wLDiag[3, -1, 5, 3],
wLDiag[3, 1, -5, -1], wLDiag[3, 1, 5, -1], wLDiag[3, 1, 5, 1], wLDiag[3, 1, 5, 3],
wLDiag[3, 5, -2, -1], wLDiag[3, 5, 2, -1], wLDiag[3, 5, 2, 1], wLDiag[4, -5, -1, -1],
wLDiag[4, -1, -5, 1], wLDiag[4, -1, -5, 2], wLDiag[4, -1, 5, 1],
wLDiag[4, 1, -5, -1], wLDiag[4, 1, 5, -1], wLDiag[4, 5, -1, -1], wLDiag[4, 5, 1, 1],
wLDiag[5, -1, -2, -3], wLDiag[5, -1, -2, -2], wLDiag[5, -1, -2, 3],
wLDiag[5, -1, -1, -3], wLDiag[5, -1, -1, -2], wLDiag[5, -1, -1, -1],
wLDiag[5, -1, -1, 3], wLDiag[5, -1, 1, -3], wLDiag[5, -1, 2, -3], wLDiag[5, -1, 2, 3],
wLDiag[5, -1, 5, -1], wLDiag[5, 1, -2, -3], wLDiag[5, 1, -2, 3], wLDiag[5, 1, 1, -3],
wLDiag[5, 1, 1, 1], wLDiag[5, 1, 1, 2], wLDiag[5, 1, 1, 3], wLDiag[5, 1, 2, -3],
wLDiag[5, 1, 2, 3], wLDiag[5, 1, 5, 1], wLDiag[5, 5, -1, -1], wLDiag[5, 5, 1, 1]}
```

AllCapped74 == AllCapped74RF

True