

Pensieve header: Finding long w-Knot diagrams all of whose rotations are non-trivial yet whose closure is trivial

```

SetDirectory["~drorbn/AcademicPensieve/Projects/wEnumeration"]
/home/drorbn/AcademicPensieve/Projects/wEnumeration

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

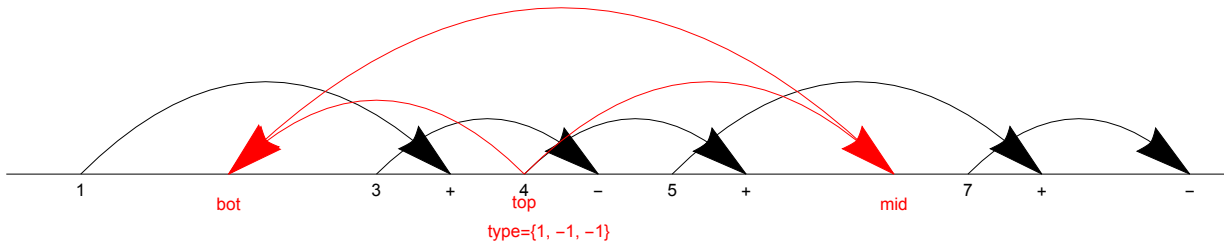
DrawArrow[i_, j_] := Arrow[BezierCurve[
  {{Abs@i, 0}, {(Abs@i + Abs@j) / 2, 0.5 Abs[Abs@i - Abs@j]}, {Abs@j, 0}}]];
Draw[wLDiag[ts___Integer] | wCDiag[ts___Integer]] := Module[{n, w, w1},
  n = Length[w = {ts}];
  w1 = Abs /@ w;
  Graphics[{
    Line[{{0, 0}, {n + 1, 0}}],
    Table[If[w[[j]] == 0, {},
      {
        DrawArrow[w1[[j]] - 0.5, j],
        Text[If[w[[j]] > 0, "+", "-"], {j, -0.1}],
        Text[w1[[j]], {w1[[j]] - 0.5, -0.1}]
      }
    ],
    {j, n}
  ]
];

Draw[wLDiag[R3[top_, mid_, bot_, s1_, s2_, s3_], ts_] |
  wCDiag[R3[top_, mid_, bot_, s1_, s2_, s3_], ts_] := Graphics[Join[
  First@Draw[wLDiag[ts]],
  {Red,
    DrawArrow[top - 0.5, mid],
    DrawArrow[top - 0.5, bot + 0.5],
    DrawArrow[mid, bot + 0.5],
    Text["bot", {bot + 0.5, -0.2}],
    Text["mid", {mid, -0.2}],
    Text["top", {top - 0.5, -0.2}],
    Text["type=" <> ToString@{s1, s2, s3}, {(top + mid + bot) / 3, -0.4}]
  }
]];

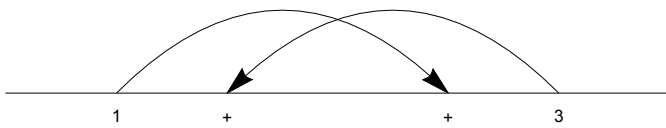
Draw[expr_] := expr /. w_wLDiag | w_wCDiag -> Draw[w]

```

```
Draw[wLDiag[R3[4, 6, 1, 1, -1, -1], 0, 0, 1, -3, 4, 0, 5, -7]]
```



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



```
AllLinearDiagrams[n_] := Flatten@Table[
  wLDiag@@@Tuples[Range[k + 1] U (-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]}
```

```

wLDiag /: RotateLeft[w_wLDiag] := Module[{n},
  n = Length[w];
  wLDiag @@ (RotateLeft[List @@ w] /. j_Integer => Which[
    j == 1, n,
    j == -1, -n,
    j > 1, j - 1,
    j < -1, j + 1
  ])
];

wCDiag /: RotateLeft[w_wCDiag] := wCDiag @@ RotateLeft[wLDiag @@ w]

RotateLeft[wCDiag[-3, 1, 3, -2]]
wCDiag[4, 2, -1, -2]

RotateToMinimal[w_wCDiag] := Module[
  {bestw = w, rotatedw = RotateLeft[w]},
  While[rotatedw != w,
    bestw = First[Sort[{bestw, rotatedw}]];
    rotatedw = RotateLeft[rotatedw]
  ];
  bestw
];

wCDiag[5, 2, -1, -2] // RotateLeft
wCDiag[1, -4, -1, 4]

NestList[RotateLeft, wCDiag[5, 2, -1, -2], 8]
{wCDiag[5, 2, -1, -2], wCDiag[1, -4, -1, 4], wCDiag[-3, -4, 3, 4],
 wCDiag[-3, 2, 3, -2], wCDiag[1, 2, -1, -2], wCDiag[1, -4, -1, 4],
 wCDiag[-3, -4, 3, 4], wCDiag[-3, 2, 3, -2], wCDiag[1, 2, -1, -2]}

wCDiag[1, -4, -1, 4] // RotateToMinimal
wCDiag[-3, -4, 3, 4]

wCDiag[w_wLDiag] := Module[{n},
  n = Length[w];
  RotateToMinimal[wCDiag @@ w /. {n + 1 -> 1, -n - 1 -> -1}]
]

AllCircularDiagrams[n_] :=
  AllCircularDiagrams[n] = Union[RotateToMinimal /@ Flatten@Table[
    wCDiag @@ Tuples[Range[k] ∪ (-Range[k]), k],
    {k, 0, n}
  ]]]

```

AllCircularDiagrams [2]

```
{wCDiag[], wCDiag[-1], wCDiag[1], wCDiag[-2, -2],
  wCDiag[-2, -1], wCDiag[-2, 1], wCDiag[-2, 2], wCDiag[-1, -2],
  wCDiag[-1, 1], wCDiag[-1, 2], wCDiag[1, 1], wCDiag[1, 2], wCDiag[2, 1]}
```

```
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]}
```

```
RemoveR1 [wCDiag[]] = wCDiag[];
```

```
RemoveR1 [w_wCDiag] := Module[{n, j, k = 0},
  n = Length[w];
  Do[If[MemberQ[{j, j + 1}, Abs[w[[j]]]], k = j], {j, n - 1}];
  If[k != 0,
    Delete[w, k] /. j_Integer /; Abs[j] > k => Sign[j] (Abs[j] - 1),
    (*else*) If[!MemberQ[{1, n}, Abs[Last[w]]], w,
      Drop[w, -1] /. {n -> 1, -n -> -1}
    ]
  ]
]
```

RemoveR1 /@AllCircularDiagrams [2]

```
{wCDiag[], wCDiag[], wCDiag[], wCDiag[-1], wCDiag[-1], wCDiag[1], wCDiag[1],
  wCDiag[-1], wCDiag[1], wCDiag[1], wCDiag[1], wCDiag[1], wCDiag[1]}
```

```
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]}
```

```
RemoveR1s /@AllCircularDiagrams[4] // Union
```

```
{wCDiag[], wCDiag[-3, -1, -2], wCDiag[-3, -1, 2], wCDiag[-3, 1, -2],
  wCDiag[-3, 1, 2], wCDiag[3, 1, 2], wCDiag[-4, -4, -2, -3], wCDiag[-4, -4, -2, -2],
  wCDiag[-4, -4, -2, 2], wCDiag[-4, -4, -2, 3], wCDiag[-4, -4, -1, -3],
  wCDiag[-4, -4, -1, -2], wCDiag[-4, -4, -1, 2], wCDiag[-4, -4, -1, 3],
  wCDiag[-4, -4, 1, -3], wCDiag[-4, -4, 1, -2], wCDiag[-4, -4, 1, 2],
  wCDiag[-4, -4, 1, 3], wCDiag[-4, -4, 2, -3], wCDiag[-4, -4, 2, -2],
  wCDiag[-4, -4, 2, 2], wCDiag[-4, -4, 2, 3], wCDiag[-4, -1, -2, -3],
  wCDiag[-4, -1, -2, 2], wCDiag[-4, -1, -2, 3], wCDiag[-4, -1, 1, -2],
  wCDiag[-4, -1, 1, 2], wCDiag[-4, -1, 1, 3], wCDiag[-4, -1, 2, -2],
  wCDiag[-4, -1, 2, 2], wCDiag[-4, -1, 2, 3], wCDiag[-4, 1, -2, 2],
  wCDiag[-4, 1, -2, 3], wCDiag[-4, 1, -1, -2], wCDiag[-4, 1, -1, 2],
  wCDiag[-4, 1, -1, 3], wCDiag[-4, 1, 1, -2], wCDiag[-4, 1, 1, 2],
  wCDiag[-4, 1, 1, 3], wCDiag[-4, 1, 2, -2], wCDiag[-4, 1, 2, 2], wCDiag[-4, 1, 2, 3],
  wCDiag[-4, 4, -2, 2], wCDiag[-4, 4, -1, -2], wCDiag[-4, 4, -1, 2],
  wCDiag[-4, 4, -1, 3], wCDiag[-4, 4, 1, -2], wCDiag[-4, 4, 1, 2],
  wCDiag[-4, 4, 1, 3], wCDiag[-4, 4, 2, -2], wCDiag[-4, 4, 2, 2], wCDiag[-4, 4, 2, 3],
  wCDiag[-3, -4, -1, -2], wCDiag[-3, -4, -1, 2], wCDiag[-3, -4, -1, 3],
  wCDiag[-3, -4, 1, 2], wCDiag[-3, -4, 1, 3], wCDiag[-3, -4, 2, 2],
  wCDiag[-3, -4, 2, 3], wCDiag[-3, 1, -1, 2], wCDiag[-3, 1, -1, 3],
  wCDiag[-3, 1, 1, 2], wCDiag[-3, 1, 1, 3], wCDiag[-3, 1, 2, 2], wCDiag[-3, 1, 2, 3],
  wCDiag[-3, 4, -1, 2], wCDiag[-3, 4, 1, 2], wCDiag[-3, 4, 1, 3],
  wCDiag[-3, 4, 2, 2], wCDiag[-3, 4, 2, 3], wCDiag[3, 1, 1, 2], wCDiag[3, 1, 1, 3],
  wCDiag[3, 1, 2, 2], wCDiag[3, 1, 2, 3], wCDiag[3, 4, 1, 2], wCDiag[4, 1, 2, 3]}
```

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

```
  Do[
```

```
    If[w[[j]] + w[[j + 1]] == 0 && !MemberQ[Abs[List@@w], j + 1], {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]}
```

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

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

RemoveR2[**w_wCDiag**] /; **Length**[**w**] < 2 := **w**;

RemoveR2[**w_wCDiag**] := **Module**[{**n**, **j**, **k** = 0},

n = **Length**[**w**];

Do[**If**[**w**[[**j**]] + **w**[[**j** + 1]] == 0 && !**MemberQ**[**Abs**[**List**@**w**], **j** + 1], **k** = **j**], {**j**, **n** - 1}];

If[**k** ≠ 0,

Delete[**w**, {{**k**}, {**k** + 1}}] /.

j_Integer /; **Abs**[**j**] > **k** => **Sign**[**j**] (**Abs**[**j**] - 2) /. {**n** - 1 → 1, 1 - **n** → -1},

If[**w**[[1]] + **w**[[**n**]] == 0 && !**MemberQ**[**Abs**[**List**@**w**], 1],

w[[2 ;; **n** - 1]] /. **j_Integer** => **Sign**[**j**] (**Abs**[**j**] - 1) /. {**n** - 1 → 1, 1 - **n** → -1},

(*else*) **w**

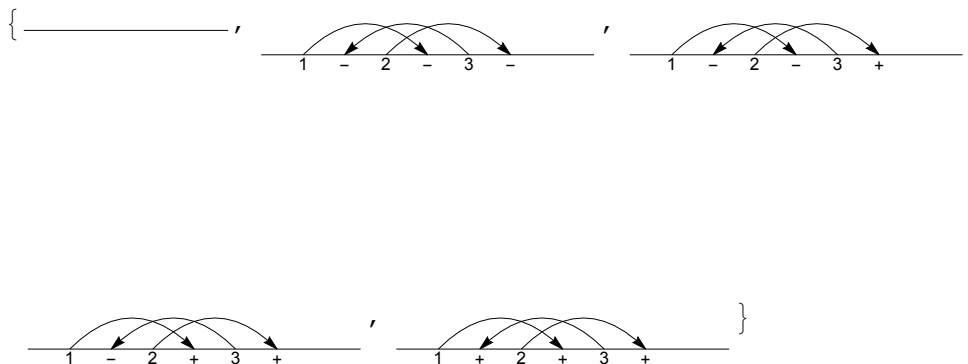
```
]
```

```
]
```

```
]
```

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

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

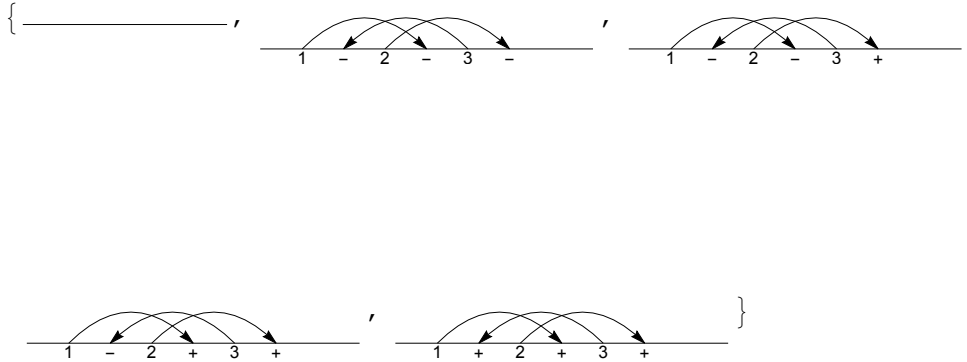


```

RF[w_wCDiag] := RF[w] = RotateToMinimal[RemoveR12s[w]];
RF[w_wLDiag] := RemoveR12s[w];

Union[RF /@ AllCircularDiagrams[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}]
  ];

```

```

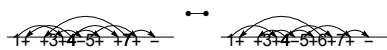
wCDiag /: Resolve[wCDiag[R3[top_, mid_, bot_, s1_, s2_, s3_], ts_]] :=
  (RF[wCDiag[#]]) & /@ Resolve[wLDiag[R3[top, mid, bot, s1, s2, s3], ts]]

```

```

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

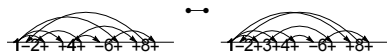
```



```

Resolve@wCDiag[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}}
];

AllCircularR3s[n_] /; n < 3 := {};
AllCircularR3s[n_] := Flatten@Table[
  Prepend[
    ReplacePart[wCDiag@@Table[0, {n}], Thread[Range[n] \ {1, 2, mid} → #]],
    R3[top, mid, 1, s1, s2, s3]
  ] & /@ Tuples[Range[-n, n] \ {-2, 0, 2}, n-3],
  {mid, Range[n] \ {1, 2}}, {top, Range[n] \ {2}},
  {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], ... 1168 ... ,
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...


```

wCircularK[n_] := wCircularK[n] = Module[{VS, vs, rule, ES, es},
  VS = AllCircularDiagrams[n];
  $k = 0; vs = Union[(++$k; RF[#]) & /@ VS];
  Print[Length@vs];
  rule = Dispatch[Thread[vs → Range[Length@vs]]];
  ES = AllCircularR3s[n];
  Print[Length@ES];
  $k = 0; es = Union[(++$k; Resolve[#] /. rule) & /@ ES];
  {vs, Graph[Range[Length@vs], es]}
];
wCircularK[n_, k_] := wCircularK[n, k] = Module[{vs, g, cc},
  {vs, g} = wCircularK[n];
  cc = ConnectedComponents[g];
  Select[Table[
    First@MinimalBy[vs[[#]] & /@ c, Length],
    {c, cc}
  ], Length[#] == k &]
]
wCircularK[4]

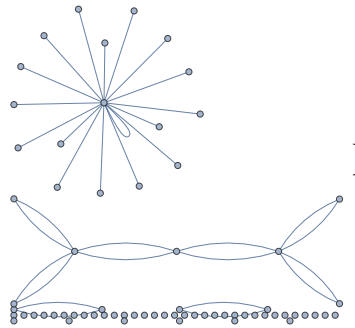
```

67

288

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

```
wCDiag[3, 4, 1, 2], wCDiag[4, 1, 2, 3]],
```



wCircularK[5, 4]

1387

6144

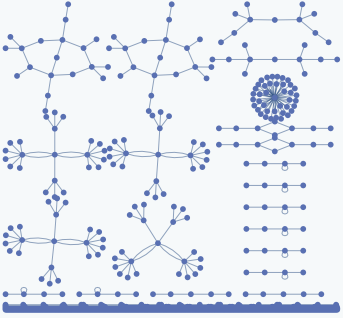
```
{wCDiag[-4, -1, -2, 3], wCDiag[-4, -4, -1, -2], wCDiag[3, 1, 2, 2],
wCDiag[-4, 4, 2, -2], wCDiag[-4, 1, 1, 2], wCDiag[-3, -4, 2, 2],
wCDiag[-4, -4, 1, 2], wCDiag[-4, -4, -1, 3], wCDiag[-4, 4, -2, 2],
wCDiag[-3, 1, 1, 2], wCDiag[-4, -4, -1, 2], wCDiag[-4, -4, 2, -3],
wCDiag[-3, 1, -1, 3], wCDiag[-4, 1, 2, 2], wCDiag[-4, -4, 1, -2],
wCDiag[-4, -4, 1, 3], wCDiag[-4, 1, 1, 3], wCDiag[-4, 1, 1, -2], wCDiag[-3, 4, 2, 2],
wCDiag[-4, -4, -2, 3], wCDiag[-4, 1, -2, 3], wCDiag[-4, -1, -2, -3],
wCDiag[3, 4, 1, 2], wCDiag[-3, -4, -1, -2], wCDiag[4, 1, 2, 3]}
```

```
wLinearK[n_] := wLinearK[n] = Module[{VS, vs, rule, ES, es},
  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[4]
```

1177

1536

```
{wLdiag[], wLdiag[-3, -1], wLdiag[-3, 1], wLdiag[3, -1],
  wLdiag[3, 1], ... 1168 ..., 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[4, 3]

```
{wLdiag[-3, -4, 2], wLdiag[3, -1, 2], wLdiag[3, -4, -2],
  wLdiag[-3, 4, 2], wLdiag[-3, -1, -2], wLdiag[3, 4, 2], wLdiag[-4, -1, 2],
  wLdiag[3, -4, -1], wLdiag[-4, 1, -2], wLdiag[3, -4, 1], wLdiag[-4, -1, -1],
  wLdiag[4, 1, 1], wLdiag[-4, 1, 2], wLdiag[-3, -4, -1], wLdiag[3, 4, 1],
  wLdiag[4, 4, -1], wLdiag[4, -1, -1], wLdiag[3, 4, -1], wLdiag[-4, 1, 1],
  wLdiag[-4, -1, -2], wLdiag[4, -1, -2], wLdiag[-4, -4, -1],
  wLdiag[-4, -4, 1], wLdiag[4, 1, 2], wLdiag[-3, -4, 1], wLdiag[4, 4, 1]}
```

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

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

```
wCDiag[3, 4, 1, 2]
```

```
ReduceMod[{vs_, g_}] := ReduceMod[{vs, g}] = Module[{cc, md},
```

```
  Dispatch[Flatten@Table[
    md = First@MinimalBy[vs[[#]] & /@ cc, Length];
    (vs[[#]] -> md) & /@ cc,
    {cc, ConnectedComponents[g]}
  ]]
```

```
ReduceMod[w_, {vs_, g_}] := (w /. ReduceMod[{vs, g}])
```

```
ReduceMod[wCDiag[3, 4, 1, -2] // RF, wCircularK[5]]
```

1387

6144

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

```
ReduceMod[RF[wLdiag@@#], wLinearK[4]] & /@ AllCircularDiagrams[3]
```

```
1177
```

```
1536
```

```
{wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[-3, -1, -2], wLdiag[-3, -1], wLdiag[3, -4, -2], wLdiag[-3, -1],
wLdiag[-3, 1], wLdiag[-3, 1], wLdiag[-3, 1], wLdiag[-3, -4, 2], wLdiag[-3, 1],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[], wLdiag[],
wLdiag[3, 4, 2]}
```

```
VeryNonTrivialQ[diag_wCdiag, sp_] :=
And@@((ReduceMod[RF[#], sp] != wLdiag[]) & /@
NestList[RotateLeft, wLdiag@@diag, Max[0, Length[diag] - 1]])
```

```
VeryNonTrivialQ[wCdiag[-4, -1, -2, 3], wLinearK[4]]
```

```
True
```

```
VeryNonTrivialQ[wCdiag[-4, -1, -1, -3], wLinearK[5]]
```

```
29129
```

```
48000
```

```
True
```

```
VeryNonTrivialQ[#, wLinearK[5]] & /@ AllCircularDiagrams[3]
```

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

```
candi3 = Select[AllCircularDiagrams[3], VeryNonTrivialQ[#, wLinearK[5]] &
```

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

```
ReduceMod[#, wCircularK[5]] & /@ candi3
```

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

```

candi4 = Select[AllCircularDiagrams[4], VeryNonTrivialQ[#, wLinearK[5]] &]
{wCDiag[-3, -1, -2], wCDiag[-3, -1, 2], wCDiag[-3, 1, 2], wCDiag[3, 1, 2],
  wCDiag[-4, -4, -2, -3], wCDiag[-4, -4, -2, -2], wCDiag[-4, -4, -2, 2],
  wCDiag[-4, -4, -2, 3], wCDiag[-4, -4, -1, -3], wCDiag[-4, -4, -1, -2],
  wCDiag[-4, -4, -1, 2], wCDiag[-4, -4, -1, 3], wCDiag[-4, -4, 1, -3],
  wCDiag[-4, -4, 1, -2], wCDiag[-4, -4, 1, 2], wCDiag[-4, -4, 1, 3],
  wCDiag[-4, -4, 2, -3], wCDiag[-4, -4, 2, -2], wCDiag[-4, -4, 2, 2],
  wCDiag[-4, -4, 2, 3], wCDiag[-4, -3, -1, -3], wCDiag[-4, -3, -1, -2],
  wCDiag[-4, -3, -1, 2], wCDiag[-4, -3, -1, 3], wCDiag[-4, -3, 1, -3],
  wCDiag[-4, -3, 1, -2], wCDiag[-4, -3, 1, 2], wCDiag[-4, -3, 1, 3],
  wCDiag[-4, -2, -1, -3], wCDiag[-4, -2, -1, -2], wCDiag[-4, -2, -1, 2],
  wCDiag[-4, -2, -1, 3], wCDiag[-4, -2, 1, -3], wCDiag[-4, -2, 1, -2],
  wCDiag[-4, -2, 1, 2], wCDiag[-4, -2, 1, 3], wCDiag[-4, -1, -2, -3],
  wCDiag[-4, -1, -2, 3], wCDiag[-4, -1, 2, 2], wCDiag[-4, -1, 2, 3],
  wCDiag[-4, -1, 3, -2], wCDiag[-4, -1, 3, 2], wCDiag[-4, -1, 4, -2],
  wCDiag[-4, -1, 4, 2], wCDiag[-4, 1, -4, -2], wCDiag[-4, 1, -4, 2],
  wCDiag[-4, 1, -3, -2], wCDiag[-4, 1, -3, 2], wCDiag[-4, 1, -2, 3],
  wCDiag[-4, 1, 1, -2], wCDiag[-4, 1, 1, 2], wCDiag[-4, 1, 1, 3], wCDiag[-4, 1, 2, 2],
  wCDiag[-4, 1, 2, 3], wCDiag[-4, 1, 3, -2], wCDiag[-4, 1, 3, 2], wCDiag[-4, 1, 4, -2],
  wCDiag[-4, 1, 4, 2], wCDiag[-4, 2, -1, -2], wCDiag[-4, 2, -1, 2],
  wCDiag[-4, 2, -1, 3], wCDiag[-4, 2, 1, -2], wCDiag[-4, 2, 1, 2], wCDiag[-4, 2, 1, 3],
  wCDiag[-4, 3, -1, -2], wCDiag[-4, 3, -1, 2], wCDiag[-4, 3, -1, 3],
  wCDiag[-4, 3, 1, -2], wCDiag[-4, 3, 1, 2], wCDiag[-4, 3, 1, 3], wCDiag[-4, 4, -2, 2],
  wCDiag[-4, 4, 2, -2], wCDiag[-4, 4, 2, 2], wCDiag[-3, -4, -1, -2],
  wCDiag[-3, -4, -1, 2], wCDiag[-3, -4, 1, 2], wCDiag[-3, -4, 2, -4],
  wCDiag[-3, -4, 2, -1], wCDiag[-3, -4, 2, 1], wCDiag[-3, -4, 2, 2],
  wCDiag[-3, -4, 2, 4], wCDiag[-3, 1, -4, 2], wCDiag[-3, 1, -3, 2],
  wCDiag[-3, 1, -1, 3], wCDiag[-3, 1, 1, 2], wCDiag[-3, 1, 1, 3], wCDiag[-3, 1, 2, -4],
  wCDiag[-3, 1, 2, -1], wCDiag[-3, 1, 2, 1], wCDiag[-3, 1, 2, 2], wCDiag[-3, 1, 2, 4],
  wCDiag[-3, 1, 3, 2], wCDiag[-3, 1, 4, 2], wCDiag[-3, 4, -1, 2], wCDiag[-3, 4, 1, 2],
  wCDiag[-3, 4, 2, -4], wCDiag[-3, 4, 2, -1], wCDiag[-3, 4, 2, 1],
  wCDiag[-3, 4, 2, 2], wCDiag[-3, 4, 2, 4], wCDiag[-2, 4, 1, 3], wCDiag[-2, 4, 2, 3],
  wCDiag[-1, 4, 1, 3], wCDiag[-1, 4, 2, 3], wCDiag[1, 4, 1, 3], wCDiag[1, 4, 2, 3],
  wCDiag[2, 4, 1, 3], wCDiag[2, 4, 2, 3], wCDiag[3, 1, 1, 2], wCDiag[3, 1, 1, 3],
  wCDiag[3, 1, 2, 2], wCDiag[3, 1, 2, 3], wCDiag[3, 4, 1, 2], wCDiag[4, 1, 2, 3]}

```

```
Union[ReduceMod[RF[#], wCircularK[5]] & /@ candi4]
```

```
{wCDiag[-3, -1, -2], wCDiag[-3, 1, 2], wCDiag[3, 1, 2],
 wCDiag[-4, -4, -2, 3], wCDiag[-4, -4, -1, -2], wCDiag[-4, -4, -1, 2],
 wCDiag[-4, -4, -1, 3], wCDiag[-4, -4, 1, -2], wCDiag[-4, -4, 1, 2],
 wCDiag[-4, -4, 1, 3], wCDiag[-4, -4, 2, -3], wCDiag[-4, -1, -2, -3],
 wCDiag[-4, -1, -2, 3], wCDiag[-4, 1, -2, 3], wCDiag[-4, 1, 1, -2],
 wCDiag[-4, 1, 1, 2], wCDiag[-4, 1, 1, 3], wCDiag[-4, 1, 2, 2],
 wCDiag[-4, 4, -2, 2], wCDiag[-4, 4, 2, -2], wCDiag[-3, -4, -1, -2],
 wCDiag[-3, -4, 2, 2], wCDiag[-3, 1, -1, 3], wCDiag[-3, 1, 1, 2],
 wCDiag[-3, 4, 2, 2], wCDiag[3, 1, 2, 2], wCDiag[3, 4, 1, 2], wCDiag[4, 1, 2, 3]}
```

```
candi5 = Select[AllCircularDiagrams[5], VeryNonTrivialQ[#], wLinearK[6]] &
```

```
877 629
```

```
1 658 880
```

```
{wCDiag[-3, -1, -2], wCDiag[-3, -1, 2], wCDiag[-3, 1, 2],
 wCDiag[3, 1, 2], wCDiag[-4, -4, -2, -3], wCDiag[-4, -4, -2, -2],
 wCDiag[-4, -4, -2, 2], wCDiag[-4, -4, -2, 3], wCDiag[-4, -4, -1, -3],
 wCDiag[-4, -4, -1, -2], ... 3926 ..., wCDiag[3, 5, 2, 3, 3],
 wCDiag[3, 5, 2, 3, 4], wCDiag[4, 1, 1, 2, 3], wCDiag[4, 1, 1, 2, 4],
 wCDiag[4, 1, 1, 3, 4], wCDiag[4, 1, 2, 2, 3], wCDiag[4, 1, 2, 3, 3],
 wCDiag[4, 1, 2, 3, 4], wCDiag[4, 5, 1, 2, 3], wCDiag[5, 1, 2, 3, 4]}
```

large output

show less

show more

show all

set size limit...

```

Select[candi5, ReduceMod[RF[#], wCircularK[6]] == wCDiag[] &]
{wCDiag[-5, -5, -2, -3, 2], wCDiag[-5, -5, -2, 1, 4],
 wCDiag[-5, -5, -2, 3, -2], wCDiag[-5, -5, 2, -3, 2], wCDiag[-5, -5, 2, -1, 4],
 wCDiag[-5, -5, 2, 1, -4], wCDiag[-5, -5, 2, 3, -2], wCDiag[-5, -4, -5, -3, 4],
 wCDiag[-5, -4, -5, 1, 4], wCDiag[-5, -4, -5, 2, -2], wCDiag[-5, -4, -2, -3, 3],
 wCDiag[-5, -4, -2, -3, 4], wCDiag[-5, -4, -2, 2, -4],
 wCDiag[-5, -4, -2, 2, 4], wCDiag[-5, -4, 1, -1, -2], wCDiag[-5, -4, 1, 1, 2],
 wCDiag[-5, -4, 2, -2, 4], wCDiag[-5, -4, 2, 3, 3], wCDiag[-5, -4, 5, -3, 4],
 wCDiag[-5, -4, 5, -2, 2], wCDiag[-5, -4, 5, -1, -4], wCDiag[-5, -4, 5, -1, -2],
 wCDiag[-5, -4, 5, -1, 4], wCDiag[-5, -4, 5, 2, -2], wCDiag[-5, -4, 5, 3, -4],
 wCDiag[-5, -1, -2, -3, 3], wCDiag[-5, -1, -2, 2, 4], wCDiag[-5, -1, 1, -3, 2],
 wCDiag[-5, -1, 1, -3, 4], wCDiag[-5, -1, 1, 3, -2], wCDiag[-5, -1, 2, -3, 2],
 wCDiag[-5, -1, 2, -1, 2], wCDiag[-5, -1, 2, 3, -2], wCDiag[-5, -1, 5, -2, 2],
 wCDiag[-5, -1, 5, -1, 2], wCDiag[-5, -1, 5, 1, -2], wCDiag[-5, -1, 5, 1, 4],
 wCDiag[-5, -1, 5, 2, -2], wCDiag[-5, -1, 5, 3, 3], wCDiag[-5, -1, 5, 3, 4],
 wCDiag[-5, 1, -5, -1, 4], wCDiag[-5, 1, -5, 1, 2], wCDiag[-5, 1, -5, 3, -3],
 wCDiag[-5, 1, -5, 3, 3], wCDiag[-5, 1, -5, 3, 4], wCDiag[-5, 1, -2, 1, 2],
 wCDiag[-5, 1, 1, -3, 2], wCDiag[-5, 1, 1, 3, -2], wCDiag[-5, 1, 2, -2, 4],
 wCDiag[-5, 1, 2, -1, -2], wCDiag[-5, 1, 2, 1, -2], wCDiag[-5, 1, 2, 3, -3],
 wCDiag[-5, 1, 5, -3, 3], wCDiag[-5, 1, 5, -1, 2], wCDiag[-5, 1, 5, -1, 4],
 wCDiag[-5, 1, 5, 1, -2], wCDiag[-5, 1, 5, 1, 4], wCDiag[-5, 1, 5, 3, -3],
 wCDiag[-5, 1, 5, 3, 4], wCDiag[-5, 4, -5, -2, -2], wCDiag[-5, 4, -5, -2, 2],
 wCDiag[-5, 4, -5, -1, 4], wCDiag[-5, 4, -5, 1, 2], wCDiag[-5, 4, -5, 2, 2],
 wCDiag[-5, 4, -5, 3, 4], wCDiag[-5, 4, -2, 2, 4], wCDiag[-5, 4, -1, -1, 2],
 wCDiag[-5, 4, -1, 1, 2], wCDiag[-5, 4, 1, -1, -2], wCDiag[-5, 4, 1, 1, -2],
 wCDiag[-5, 4, 2, 2, 4], wCDiag[-5, 4, 2, 3, -3], wCDiag[-5, 4, 5, -3, 3],
 wCDiag[-5, 4, 5, -2, -2], wCDiag[-5, 4, 5, -1, -2], wCDiag[-5, 4, 5, 2, 2],
 wCDiag[-5, 4, 5, 3, -3], wCDiag[-5, 5, 2, -1, -2], wCDiag[-4, -4, -5, 3, 2],
 wCDiag[-4, -4, 2, -1, -2], wCDiag[-4, -4, 2, -1, 2], wCDiag[-4, -4, 5, 3, -2],
 wCDiag[-4, 1, -5, -1, 4], wCDiag[-4, 1, -5, 3, 4], wCDiag[-4, 1, 2, 1, 4],
 wCDiag[-4, 1, 2, 3, 4], wCDiag[-4, 1, 5, 3, 4], wCDiag[-4, 4, 2, 1, -2],
 wCDiag[-4, 4, 2, 1, 2], wCDiag[-4, 4, 2, 3, -2], wCDiag[-4, 4, 5, -1, 2],
 wCDiag[-4, 4, 5, 1, -2], wCDiag[-4, 4, 5, 1, 2], wCDiag[-4, 4, 5, 3, -2],
 wCDiag[-4, 4, 5, 3, 2], wCDiag[-3, -4, -5, -1, 3], wCDiag[-3, -4, -5, 1, 3],
 wCDiag[-3, -4, 1, 1, 4], wCDiag[-3, -4, 2, -1, 2], wCDiag[-3, -4, 2, -1, 3],
 wCDiag[-3, -4, 5, -1, 3], wCDiag[-3, -4, 5, 1, 4], wCDiag[-3, -4, 5, 3, 4],
 wCDiag[-3, 1, -5, 1, 3], wCDiag[-3, 1, -5, 2, 2], wCDiag[-3, 1, 1, 3, 4],
 wCDiag[-3, 1, 2, -1, 2], wCDiag[-3, 1, 2, -1, 3], wCDiag[-3, 1, 5, -1, 3],
 wCDiag[-3, 4, 1, 1, 4], wCDiag[-3, 4, 2, 3, 2], wCDiag[-3, 4, 5, 3, 2]}

candi5in7 = Select[AllCircularDiagrams[5], VeryNonTrivialQ[#, wLinearK[7]] &]

```