Pensieve header: Counting virtual pure braids: Cartesian product diagrams immediately serialized. May contain a Mathematica- or an OS- or a Hardware- induced crash in the computation of CountVPB[5,{4,6}].

An ERO is an Equivalence Relation Object, as in EquivalenceRelations.nb.

(Alt) In[•]:=
```
SetAttributes[{EROMake, EROPeek, EROAdjoin}, HoldFirst];
```

(Alt) In[•]:=
```
EROMake[er_, n_Integer] := er = Table[0, n];
```

(Alt) In[•]:=
```
EROPeek[er_, n_Integer] := If[er[[n]] == 0, n, er[[n]] = EROPeek[er, er[[n]]]];
```

(Alt) In[•]:=
```
EROAdjoin[er_, new_UndirectedEdge] := EROAdjoin[er, new, Identity];
EROAdjoin[er_, n1_Integer ⟞ n2_Integer, comp_] := Module[{m1, m2},
  m1 = EROPeek[er, n1]; m2 = EROPeek[er, n2];
  Switch[Order[{comp[m1], m1}, {comp[m2], m2}],
   0, m1, 1, er[[m2]] = m1, -1, er[[m1]] = m2] ]
```

(Alt) In[•]:=
```
VPB[n_, gs_List] := 1 + Sum[(2 n (n - 1))^s, {s, 0, Length[gs] - 1}] + FromDigits[gs //.
     {σ_i_,j_ :> (n - 1) (i - 1) + If[j < i, j - 1, j - 2], σ̄_i_,j_ :> n (n - 1) + σ_i,j}, 2 n (n - 1)];
```

(Alt) In[•]:=
```
VPB[n_, c_Integer] := Module[{c1, cc, r = 0, s, i, j, d},
  c1 = cc = c - 1;
  While[(c1 = cc - (2 n (n - 1))^r) ≥ 0, cc = c1; ++r];
  Table[
   {r, i, j} = 1 + IntegerDigits[d, MixedRadix[{2, n, n - 1}], 3];
   If[j ≥ i, ++j];
   If[r == 1, σ_i,j, σ̄_i,j],
   {d, IntegerDigits[cc, 2 n (n - 1), r]}
  ]]
```

In[•]:=
```
VPB[3, #] & /@ Range[50]
```

Out[•]=
$$\{\{\},\ \{\sigma_{1,2}\},\ \{\sigma_{1,3}\},\ \{\sigma_{2,1}\},\ \{\sigma_{2,3}\},\ \{\sigma_{3,1}\},\ \{\sigma_{3,2}\},\ \{\bar\sigma_{1,2}\},\ \{\bar\sigma_{1,3}\},\ \{\bar\sigma_{2,1}\},\ \{\bar\sigma_{2,3}\},\ \{\bar\sigma_{3,1}\},$$
$$\{\bar\sigma_{3,2}\},\ \{\sigma_{1,2},\sigma_{1,2}\},\ \{\sigma_{1,2},\sigma_{1,3}\},\ \{\sigma_{1,2},\sigma_{2,1}\},\ \{\sigma_{1,2},\sigma_{2,3}\},\ \{\sigma_{1,2},\sigma_{3,1}\},\ \{\sigma_{1,2},\sigma_{3,2}\},$$
$$\{\sigma_{1,2},\bar\sigma_{1,2}\},\ \{\sigma_{1,2},\bar\sigma_{1,3}\},\ \{\sigma_{1,2},\bar\sigma_{2,1}\},\ \{\sigma_{1,2},\bar\sigma_{2,3}\},\ \{\sigma_{1,2},\bar\sigma_{3,1}\},\ \{\sigma_{1,2},\bar\sigma_{3,2}\},\ \{\sigma_{1,3},\sigma_{1,2}\},$$
$$\{\sigma_{1,3},\sigma_{1,3}\},\ \{\sigma_{1,3},\sigma_{2,1}\},\ \{\sigma_{1,3},\sigma_{2,3}\},\ \{\sigma_{1,3},\sigma_{3,1}\},\ \{\sigma_{1,3},\sigma_{3,2}\},\ \{\sigma_{1,3},\bar\sigma_{1,2}\},$$
$$\{\sigma_{1,3},\bar\sigma_{1,3}\},\ \{\sigma_{1,3},\bar\sigma_{2,1}\},\ \{\sigma_{1,3},\bar\sigma_{2,3}\},\ \{\sigma_{1,3},\bar\sigma_{3,1}\},\ \{\sigma_{1,3},\bar\sigma_{3,2}\},\ \{\sigma_{2,1},\sigma_{1,2}\},$$
$$\{\sigma_{2,1},\sigma_{1,3}\},\ \{\sigma_{2,1},\sigma_{2,1}\},\ \{\sigma_{2,1},\sigma_{2,3}\},\ \{\sigma_{2,1},\sigma_{3,1}\},\ \{\sigma_{2,1},\sigma_{3,2}\},\ \{\sigma_{2,1},\bar\sigma_{1,2}\},$$
$$\{\sigma_{2,1},\bar\sigma_{1,3}\},\ \{\sigma_{2,1},\bar\sigma_{2,1}\},\ \{\sigma_{2,1},\bar\sigma_{2,3}\},\ \{\sigma_{2,1},\bar\sigma_{3,1}\},\ \{\sigma_{2,1},\bar\sigma_{3,2}\},\ \{\sigma_{2,3},\sigma_{1,2}\}\}$$

In[•]:=
```
Range[50] === (VPB[3, #] & /@ (VPB[3, #] & /@ Range[50]))
```

Out[•]=
```
True
```

(Alt) In[ ]:=

```
CountVPB[n_, m_] := CountVPB[n, {m, m}];
CountVPB[n_, {m1_, m2_}] :=
 Module[{σ, gens, dc, d2n, s, VPB, T, ij, ijk, ijkl, i, j, k, l, perm},
  {σ_i_,j_ := (n - 1) (i - 1) + If[j < i, j - 1, j - 2], σ̄_i_,j_ := n (n - 1) + σ_i,j};
  gens = Range[2 n (n - 1)] - 1;
  dc[m_] := dc[m] = Sum[(2 n (n - 1))^s, {s, 0, m}];
  Print[dc /@ {m1, m2}, " diagrams..."];
  EROMake[$er, dc[m2]];
  VPB[_, gs_List] := 1 + dc[Length[gs] - 1] + FromDigits[gs, 2 n (n - 1)];
  T[b1_, b2_] := EROAdjoin[$er, b1 ⟼ b2];
  Do[{i, j} = ij; {
    T[VPB[n, Join[p, {σ_i,j, σ̄_i,j}, q]], VPB[n, Join[p, q]]],
    T[VPB[n, Join[p, {σ̄_i,j, σ_i,j}, q]], VPB[n, Join[p, q]]]
   },
   {s, 0, m2 - 2}, {t, 0, s}, {p, Tuples[gens, t]}, {q, Tuples[gens, s - t]},
   {ij, Join @@ (Permutations /@ Subsets[Range[n], {2}])}
  ];
  Do[{i, j, k} = ijk; {
    T[VPB[n, Join[p, {σ_i,j, σ_i,k, σ_j,k}, q]], VPB[n, Join[p, {σ_j,k, σ_i,k, σ_i,j}, q]]],
    T[VPB[n, Join[p, {σ̄_j,i, σ_i,k, σ_j,k}, q]], VPB[n, Join[p, {σ_j,k, σ_i,k, σ̄_j,i}, q]]],
    T[VPB[n, Join[p, {σ_i,j, σ_i,k, σ̄_k,j}, q]], VPB[n, Join[p, {σ̄_k,j, σ_i,k, σ_i,j}, q]]],
    T[VPB[n, Join[p, {σ_i,j, σ̄_k,i, σ̄_k,j}, q]], VPB[n, Join[p, {σ̄_k,j, σ̄_k,i, σ_i,j}, q]]],
    T[VPB[n, Join[p, {σ̄_j,i, σ̄_k,i, σ_j,k}, q]], VPB[n, Join[p, {σ_j,k, σ̄_k,i, σ̄_j,i}, q]]],
    T[VPB[n, Join[p, {σ̄_j,i, σ̄_k,i, σ̄_k,j}, q]], VPB[n, Join[p, {σ̄_k,j, σ̄_k,i, σ̄_j,i}, q]]]
   },
   {s, 0, m2 - 3}, {t, 0, s}, {p, Tuples[gens, t]}, {q, Tuples[gens, s - t]},
   {ijk, Join @@ (Permutations /@ Subsets[Range[n], {3}])}
  ];
  Do[{i, j, k, l} = ijkl; {
    T[VPB[n, Join[p, {σ_i,j, σ_k,l}, q]], VPB[n, Join[p, {σ_k,l, σ_i,j}, q]]],
    T[VPB[n, Join[p, {σ̄_i,j, σ_k,l}, q]], VPB[n, Join[p, {σ_k,l, σ̄_i,j}, q]]],
    T[VPB[n, Join[p, {σ_i,j, σ̄_k,l}, q]], VPB[n, Join[p, {σ̄_k,l, σ_i,j}, q]]],
    T[VPB[n, Join[p, {σ̄_i,j, σ̄_k,l}, q]], VPB[n, Join[p, {σ̄_k,l, σ̄_i,j}, q]]]
   },
   {s, 0, m2 - 2}, {t, 0, s}, {p, Tuples[gens, t]}, {q, Tuples[gens, s - t]},
   {ijkl, Join @@ (Permutations /@ Subsets[Range[n], {4}])}
  ];
  Count[Take[$er, dc[m1]], 0]
 ]
```

In[ ]:=

```
VPB[4, {σ_4,1, σ̄_2,3}]
```

Out[ ]= 331

In[ ]:=

```
CountVPB[2, 1]
```

```
{5, 5} diagrams...
```

Out[ ]= 5

```
In[•]:= CountVPB[2, 2]
        {21, 21} diagrams...
Out[•]= 17

In[•]:= Timing@CountVPB[2, 3]
        {85, 85} diagrams...
Out[•]= {0., 53}

In[•]:= Timing@CountVPB[2, {3, 4}]
        {85, 341} diagrams...
Out[•]= {0.015625, 53}

In[•]:= Timing@CountVPB[2, 4]
        {341, 341} diagrams...
Out[•]= {0.015625, 161}

(Alt) In[•]:= Timing@CountVPB[2, 5]
        {1365, 1365} diagrams...
(Alt) Out[•]= {0.046875, 485}

(Alt) In[•]:= Timing@CountVPB[2, 6]
        {5461, 5461} diagrams...
(Alt) Out[•]= {0.25, 1457}

In[•]:= Timing@CountVPB[3, 1]
        {13, 13} diagrams...
Out[•]= {0., 13}

In[•]:= Timing@CountVPB[3, 2]
        {157, 157} diagrams...
Out[•]= {0., 145}

In[•]:= Timing@CountVPB[3, 3]
        {1885, 1885} diagrams...
Out[•]= {0.015625, 1561}

In[•]:= Timing@CountVPB[3, {3, 4}]
        {1885, 22621} diagrams...
Out[•]= {0.25, 1561}
```

In[•]:= **Timing@CountVPB[3, 4]**

{22 621, 22 621} diagrams...

Out[•]= {0.234375, 16 741}

In[•]:= **Timing@CountVPB[3, {4, 5}]**

{22 621, 271 453} diagrams...

Out[•]= {3.78125, 16 741}

In[•]:= **Timing@CountVPB[3, {4, 6}]**

{22 621, 3 257 437} diagrams...

Out[•]= {58.9063, 16 717}

In[•]:= **Timing@CountVPB[3, {4, 7}]**

{22 621, 39 089 245} diagrams...

Out[•]= {890.328, 16 717}

In[•]:= **Timing@CountVPB[3, 5]**

{271 453, 271 453} diagrams...

Out[•]= {3.92188, 179 401}

In[•]:= **Timing@CountVPB[3, {5, 6}]**

{271 453, 3 257 437} diagrams...

Out[•]= {63.7031, 179 377}

In[•]:= **Timing@CountVPB[3, {5, 7}]**

{271 453, 39 089 245} diagrams...

Out[•]= {918.016, 178 873}

In[•]:= **Timing@CountVPB[4, 1]**

{25, 25} diagrams...

Out[•]= {0., 25}

In[•]:= **Timing@CountVPB[4, 2]**

{601, 601} diagrams...

Out[•]= {0., 529}

In[•]:= **Timing@CountVPB[4, 3]**

{14 425, 14 425} diagrams...

Out[•]= {0.296875, 10 873}

In[•]:= **Timing@CountVPB[4, 4]**

{346 201, 346 201} diagrams...

Out[•]= {10.0469, 222 385}

In[•]:= **Timing@CountVPB[4, {4, 5}]**

{346 201, 8 308 825} diagrams...

Out[•]= {329.063, 222 385}

In[•]:= **Timing@CountVPB[4, {4, 6}]**

{346 201, 199 411 801} diagrams...

Out[•]= {12 658.6, 222 289}

In[•]:= **Timing@CountVPB[5, 1]**

{41, 41} diagrams...

Out[•]= {0., 41}

In[•]:= **Timing@CountVPB[5, 2]**

{1641, 1641} diagrams...

Out[•]= {0.03125, 1361}

In[•]:= **Timing@CountVPB[5, 3]**

{65 641, 65 641} diagrams...

Out[•]= {1.98438, 43 121}

(Alt) In[•]:= **Timing@CountVPB[5, 4]**

{2 625 641, 2 625 641} diagrams...

(Alt) Out[•]= {105., 1 351 721}

(Alt) In[•]:= **Timing@CountVPB[5, {4, 6}]**

{2 625 641, 4 201 025 641} diagrams...

Part: Part 68721573889 of {<<33608205376 bytes>>} does not exist.

Set: The expression If[<<33608205856 bytes>>] cannot be used as a part specification.

(Alt) In[•]:= **VPB[5, 68 721 573 889]**

(Alt) Out[•]= $\left\{\sigma_{4,5},\ \overline{\sigma}_{3,4},\ \sigma_{1,5},\ \sigma_{4,2},\ \overline{\sigma}_{1,4},\ \overline{\sigma}_{2,4},\ \sigma_{2,5}\right\}$

(Alt) In[•]:= **Timing@CountVPB[6, 1]**

{61, 61} diagrams...

(Alt) Out[•]= {0., 61}

(Alt) In[ ]:=
```
Timing@CountVPB[6, 2]
```

```
{3661, 3661} diagrams...
```

(Alt) Out[ ]=
```
{0.0625, 2881}
```

(Alt) In[ ]:=
```
Timing@CountVPB[6, 3]
```

```
{219 661, 219 661} diagrams...
```

(Alt) Out[ ]=
```
{8.53125, 127 021}
```