

Простые однотоочечные числа Гурвица брутфорсом по определению:

$$h_{m,\mu}^{\circ} = \frac{1}{n!} \left| \left\{ (\tau_1, \dots, \tau_m) \in C_2^n \mid \tau_m \circ \dots \circ \tau_1 \in C_{\mu} \right\} \right|$$

## Несвязные простые однотоочечные ЧГ по определению

```
In[*]:= IsSameCycleType[perm1_, perm2_] :=
  Equal[Sort[Tally[Map[Length, perm1[[1]]]], Sort[Tally[Map[Length, perm2[[1]]]]]
  равно сор... подс... л... длина сор... подс... л... длина
HasCycleType[perm_, part_] :=
  Equal[Sort[Tally[Map[Length, perm[[1]]]], Sort[Tally[Select[part, # != 1 &]]]]
  равно сор... подс... л... длина сор... подс... выбрать
hSimpleBruteforce[m_, part_] := Count[PermutationProduct @@@
  встр... произведение перестановок
  Map[Function[Cycles[{#}]], Tuples[Subsets[Range[Plus @@ part], {2}], m], {2}],
  л... функция циклы набор... подмно... диап... сложить
  (* Generating all m-tuples of transpositions from S_n, n=|partition| *)
  perm_ /; HasCycleType[perm, part]] / Factorial[Plus @@ part]
  факториал сложить
```

## Связные простые однотоочечные ЧГ по определению

```
In[*]:= IsTranspositionTupleTransitive[tuple_] :=
  (Length[GroupOrbits[PermutationGroup[tuple]]] == 1)
  длина орбиты группы группа перестановок
hSimpleConnectedBruteforce[m_, part_] := Count[PermutationProduct @@@
  встр... произведение перестановок
  Map[Function[Cycles[{#}]], Tuples[Subsets[Range[Plus @@ part], {2}], m], {2}],
  л... функция циклы набор... подмно... диап... сложить
  (* Generating all m-tuples of transpositions from S_n, n=|partition| *)
  perm_ /; HasCycleType[perm, part]] / Factorial[Plus @@ part]
  факториал сложить
```

## Через уравнение транспозиции

Обозначим  $A(n) = \sum_{i+j=n} (A_1(i, j) + A_2(i, j))$ , где  $A_1(i, j) = (i+j) p_i p_j \frac{\partial}{\partial p_{i+j}}$  и  $A_2(i, j) = i j p_{i+j} \frac{\partial^2}{\partial p_i \partial p_j}$ .

Оператор транспозиции:  $A = \sum_{n=1}^{\infty} A(n)$

## Простые одноточечные числа Гурвица

In[1]:= upTo := 20

```
hSimpleTransposition[m_, part_] := (
  A1[f_, {i_, j_}] := (i + j) p[i] × p[j] × D[f, p[i + j]];
  A2[f_, {i_, j_}] := i j p[i + j] × D[D[f, p[i]], p[j]];
  Asum[f_, A_, n_Integer] :=
    Plus@@ (Table[A[f, pair], {pair, Table[{x, n - x}, {x, n - 1}]}]);
  An[n_Integer, f_] := 1 / 2 (Asum[f, A1, n] + Asum[f, A2, n]);
  A[f_] := Expand[Plus@@ Table[An[x, f], {x, 1, upTo}]];
  H[nests_Integer] := A[H[nests - 1]];
  H[0] := Normal[Series[E^p[1], {p[1], 0, upTo}]];
  FreeCoefficient[c_] := c /. p[_] → 0;
  FreeCoefficient[Coefficient[H[m], Times@@ (Map[p, part])]]
)
```

## Связные одноточечные числа Гурвица

# Аглебра крашенных перестановок

Крашенная перестановка -- пара из перестановки и раскраски ее инвариантных подмножеств.  
Будем обозначать за ColoredCycles.

```
ColoredCycles[
  Cycles[{{1, 2}, {3}}], (* Permutation *)
  {{1}, {2}, {3}} (* Coloring *)
]
```

In[\*]:= MyIntersection[sets\_] := Union@@ sets

```

In[ ]:= CommonSubpartition[part1_, part2_] := (* slow but readable *) (
  n = Max[Union[part1]];
  ⌊ма... ⌋объединение
  part = Union[part1, part2];
  ⌊объединение
  CleanPart[part_, k_] := ( (* unite evetything that has i *)
    united = Union @@ Select[part, MemberQ[#, k] &];
    ⌊объеди... ⌋выбрать ⌊элемент списка?
    res = Append[Select[part, Not[MemberQ[#, k] &], united];
    ⌊добав... ⌋выбрать ⌊от... ⌋элемент списка?
    res
  );
  For[i = 1, i ≤ n, i++,
  ⌊цикл ДЛЯ
    part = CleanPart[part, i];
  ];
  part
)
part1 = {{1, 3}, {2}}
part2 = {{1, 2, 3}}
CommonSubpartition[part1, part2]

```

```
Out[ ]:= {{1, 3}, {2}}
```

```
Out[ ]:= {{1, 2, 3}}
```

```
Out[ ]:= {{1, 2, 3}}
```

```

In[ ]:= ColoredCyclesProduct[cycles1_, cycles2_] := ColoredCycles[
  PermutationProduct[cycles1[[1]], cycles2[[1]],
  ⌊произведение перестановок
  CommonSubpartition[cycles1[[2]], cycles2[[2]]]
]

```

```

In[ ]:= cycle1 = Cycles[{{1, 3}, {2}}]
  ⌊циклы
  colcycles1 = ColoredCycles[cycle1, part1]
  cycles2 = Cycles[{{1, 2, 3}}]
  ⌊циклы
  colcycles2 = ColoredCycles[cycles2, part2]

```

```
Out[ ]:= Cycles[{{1, 3}}]
```

```
Out[ ]:= ColoredCycles[Cycles[{{1, 3}}], {{1, 3}, {2}}]
```

```
Out[ ]:= Cycles[{{1, 2, 3}}]
```

```
Out[ ]:= ColoredCycles[Cycles[{{1, 2, 3}}], {{1, 2, 3}}]
```

```
In[ ]:= ColoredCyclesProduct[colcycles1, colcycles2]
```

```
Out[ ]:= ColoredCycles[Cycles[{{2, 3}}], {{1, 2, 3}}]
```