Algorithm 1: Counting monomials with only even exponents for given even partition of number n

Input : Even partition $\mathcal{P} = \{p_1, ..., p_k\}$ of the number n, problem dimension d,number of random samples m

Output: Number of monomials with only even exponents

- $S := 0$
- For every multipermutation of $\mathcal{P}$ do:
  - Create $\mathcal{T} = \{\underbrace{1, ..., 1}_{p_1}, ..., \underbrace{k, ..., k}_{p_k}\}$
  - For every multipermutation of $\mathcal{T}$ do:
    - ∗ Check if $\mathcal{T}$ is correct with Algorithm (5).
      If Algortihm (5) returned false, go to the next multipermutation of $\mathcal{P}$.
    - ∗ Create matrices D and M for the table $\mathcal{T}$ with Algorithm (2).
    - ∗ If there are unfilled places in matrices D and M, create pairs of matrices D' and M' by filling unfilled places in D and M in every possible way.
    - ∗ From the set of every generated in previous step pairs of matrices, discard these which are incorrect. Correctnes of the pair of matrices is check with Algorithm (3).
    - ∗ For every pair of matrices calculate it's numerical value with Algorithm (4) and add it to S.
- Return S

Algorithm 2: Creating matrices D and M

Input : $\mathcal{T} = \{t_1, ..., t_n\}$, where $t_i \in \{1, ..., k\}$

Output: Filled matrices D and M

- Create matrices D and M of a dimension k
- From $i = 1$ to n do:
  - If i is odd, do:
    - ∗ To $M_{t_i t_{i+1}}$ and $M_{t_{i+1} t_i}$ write "$=$".
    - ∗ To $D_{t_i t_{i+1}}$ and $D_{t_{i+1} t_i}$ write "$\neq$".
  - If i is even, do:
    - ∗ To $D_{t_i t_{i+1}}$ and $D_{t_{i+1} t_i}$ write "$=$".
    - ∗ To $M_{t_i t_{i+1}}$ and $M_{t_{i+1} t_i}$ write "$\neq$".
- Return matrices D and M

Note: If there is an occurence of writing "$\neq$" or "$=$" in previously filled place, stop the algorithm and don't return anything.

Algorithm 3: Checking if given pair of matrices D and M is correct

Input : Matrices D and M
Output: *True*, if given pair is correct; *False* in other cases

- For every pair of indexes i, j of the matrix do:
  - If $D_{ij} = M_{ij} = $ "="
    - Return *False*
- If the negation of any from implications listed below is true:
  $[(X_{ij} = \text{"="}) \wedge (X_{ik} = \text{"="})] \Rightarrow X_{kj} = \text{"="}$ or $[(X_{ij} = \text{"="}) \wedge (X_{ik} = \text{"}\neq\text{"})] \Rightarrow X_{kj} = \text{"}\neq\text{"}$
  where X is a matrix D or M
  - Return *False*
- Return *True*

Algorithm 4: Calculating a component of sum from formula for number of monomials

Input : Pair of matrices D and M, problem dimension d, number of random samples m
Output: A component of sum from formula for number of monomials

- $q = 1$
- From $i = 1$ to $\dim D$ do:
  - If $i = 1$
    - $q = q \cdot d \cdot m$
  - Else:
    - Calculate c_d oraz c_m with Algorithm (6) with input data: (D, i) and (M, i)
    - If there exists $i < j$, such that $D_{ij} = \text{"}\neq\text{"}$, do:
      - $q = q \cdot (d - c_d)$
    - If there exists $i < j$, such that $M_{ij} = \text{"}\neq\text{"}$, do:
      - $q = q \cdot (m - c_m)$
- Retrun q

Algorithm 5: Checking if table $\mathcal{T}$ is correct

Input : $\mathcal{T} = \{t_1, ..., t_n\}$, where $t_i \in \{1, ..., k\}$
Output: *True*, if $\mathcal{T}$ is correct; *False* in other cases

- $A = \emptyset$
- From $i = 1$ to n do:
  - If there exists $s < i$, such that $t_s > t_i$ and $t_i \notin A$, do:
    - Return *False*
  - insert t_i into A
- Return *True*

Algorithm 6: Calculating auxiliary minuend

Input : Matrix X, number i

Output: Auxiliary minuend

- If $i = 1$ do:
  - Return 0
- For each $j < i$ do:
  - If $X_{ij} =$ "$\neq$" do:
    - Calculate c_j with Algorithm (6) with input data (X, j)
- $c = \min_j c_j$
- Return $c + 1$