Here each of the graph is represented as a 10 bit bi-ry string where each bit represent which of the edges[(0,1), (0,2), (0,3), (0,4), (1,2), (1,3), (1,4), (2,3), (2,4), (3,4)] exists in the graph.

Nice Tree Decomposition: A tree decomposition is Nice for a 5-vertex graph, if the tree width is 2 and the vertices can be spanned by a vertex and an edge.

Conditions: The conditions helps us to reduce the automorphism to odd automorphisms which helps us in cancelling the even terms present in the equation.
For Example for G02, 0011001010, This graph has two automorphisms, corresponding to swapping vertices 1 and 2. To eliminate this symmetry, we impose the condition $1 < 2$. We then consider only those homomorphisms that respect this ordering and discard any homomorphism that violates it.

```
from itertools import combinations

def vertices_covered_by_q_edges(H, q):
    """
    Return True if there exist q edges in H
    whose endpoints cover all vertices of H.
    """
    V = set(H.vertices())
    E = list(H.edges(labels=False))

    if not V:
        return True  # empty graph

    if len(V) > 2*q:
        return False  # impossible even in best case

    for edgeset in combinations(E, q):
        covered = set()
        for u, v in edgeset:
            covered.add(u)
            covered.add(v)
        if covered == V:
            return True

    return False


def bag_is_spannable(G, bag, p, q):
    """
    Check if bag can be spanned by p vertices and q edges.
    """
    H = G.subgraph(bag)
    B = list(bag)

    if len(B) < p:
        return False

    for P in combinations(B, p):
        R = [v for v in B if v not in P]
        HR = H.subgraph(R)
        if vertices_covered_by_q_edges(HR, q):
            return True

    return False


def verify_decomposition(G, T, conditions, p, q):
    p = int(p)
    q = int(q)

    # Parse bags
    bags = T.split("-")
    bag_lists = [[int(v) for v in bag] for bag in bags]

    # Parse conditions
    conds = []
    for c in conditions.split(","):
        c = c.strip()
        if not c:
            continue
        i, j = map(int, c.split("<"))
        if i >= j:
            return False, f"Invalid condition {i}<{j}"
        conds.append((i, j))

    # Spanning condition
    for idx, bag in enumerate(bag_lists):
        if not bag_is_spannable(G, bag, p, q):
            return (
                False,
                f"Bag {idx+1} not spannable by p={p}, q={q}"
            )

    # Comparability condition
    for (i, j) in conds:
        found = False
        for bag in bag_lists:
            if i in bag and j in bag:
                found = True
                break
        if not found:
            return False, f"Condition {i}<{j} not witnessed"

    # Automorphism parity condition
    count = 0
    conds = [(str(i), str(j)) for (i, j) in conds]

    for sigma in G.automorphism_group():
        ok = True
        for (i, j) in conds:
            if sigma(i) >= sigma(j):
```

```
                ok = False
                break
        if ok:
            count += 1

    if count % 2 == 0:
        return False, f"Automorphism count even ({count})"

    return True, "OK"

G = Graph([('4','3'), ('3','0'), ('0','4'), ('4','2'), ('4','1')])

T = "043-124"
conditions = "1<2, 0<3"

verify_decomposition(G, T, conditions, p=1, q=1)
```

Verification procedure. The function `verify_decomposition` checks whether a given sequence of bags is valid for parameters p and q. Each bag must be (p, q)-spannable: after removing p vertices, the remaining vertices are covered by the endpoints of q edges. This is verified by enumerating all choices of removed vertices and testing edge coverage in the induced subgraph.

The procedure also checks that every comparability condition $i < j$ is witnessed by some bag, and that the number of graph automorphisms preserving all such conditions is odd. Any violation causes the verification to fail with an explicit error.
Below is the table of all the possible 5 vertex simple connected graphs with their (p,q) tree decomposition, we can use the above code to verify each tree decomposition is a valid tree decomposition, can be spanned by at most p vertices and q edges, and after applying conditions only odd automorphisms survive.

Name	Adjacency	#Aut	t.d.	Checks	#AutRem
G01	0001001011	24	124-024-034	-	-
G02	0011001010	2	034-142	$1 < 2$	1
G03	0011001011	4	034-124-234	$1 < 2, 2 < 3, 3 < 0$	1
G04	0011011010	2	013-014-24	$0 < 1$	1
G05	0011010011	2	013-034-24	$3 < 4$	1
G06	0011011011	2	034-134-24	$0 < 1$	1
G07	0011011110	12	034-134-234	-	-
G08	0011011111	12	034-134-234	-	-
G09	0101011000	2	042-041-13	$0 < 1$	1
G10	0101011010	2	042-134	$0 < 2$	1
G11	0101011011	8	024-034-134	$2 < 0, 0 < 3, 3 < 1$	1
G12	0110011010	10	143-034-024	$3 < 4$	5
G13	0111011010	2	024-014-013	$0 < 4$	1
G14	0111011011	2	134-034-024	$0 < 3$	1
G15	0111001110	2	14-024-023	$0 < 2$	1
G16	0111001111	6	-	-	-
G17	0111011111	4	-	-	-
G18	0111111010	4	-	-	-
G19	0111111011	8	-	-	-
G20	0111111111	12	-	-	-
G21	1111111111	120	-	-	-

[No of G_i in G_j]	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	G21
G01	1	0	1	0	0	1	0	2	0	0	1	0	0	1	0	1	2	0	1	3	5
G02	0	1	2	2	2	5	6	12	0	1	4	0	4	10	4	9	20	10	20	36	60
G03	0	0	1	0	0	2	0	6	0	0	2	0	0	3	0	3	8	0	4	15	30
G04	0	0	0	1	0	1	6	6	0	0	0	0	2	4	1	3	12	8	16	30	60
G05	0	0	0	0	1	2	0	6	0	0	0	0	1	5	2	6	14	4	12	30	60
G06	0	0	0	0	0	1	0	6	0	0	0	0	0	2	0	3	10	0	4	24	60
G07	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	2	4	10
G08	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	3	10
G09	0	0	0	2	1	2	6	6	1	2	4	5	7	10	4	6	18	14	24	36	60
G10	0	0	0	0	0	0	0	0	0	1	4	0	2	6	2	3	12	6	16	30	60
G11	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	2	0	2	6	15
G12	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	2	2	4	6	12
G13	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	6	4	12	24	60
G14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	0	4	18	60
G15	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	3	8	2	8	24	60
G16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	6	20
G17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	6	30
G18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4	9	30
G19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	15
G20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10
G21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Each of the rows are classified into 3 categories in the above table. represents all the graphs which have a valid (p,q) tree decomposition and after applying conditions only odd automorphisms survive. These graphs and graphs with no color assigned are the graphs which we can compute in $O(n, m)$ time. represents those graphs which has a $K_4 - minor$. represents those graphs which have odd number of occurrences in a $K_4 - minor$ graph.

Below is the data for the graphs in which path of length 7 occurs odd number of times.

Name	Adjacency	#Aut	t.d.	Checks	#AutRem
G001	00011100101001001010	1	01346-025	-	1
G002	00011100100001100101 0	1	01346-02456	-	1
G003	00011000101001000101 1	1	1346-02456	-	1
G004	00011100101001000101 1	1	01346-0256	-	1
G005	00010100100001101001 0	1	235-01246	-	1
G006	00011100110001100101 0	1	1356-02456	-	1
G007	00011100110001100101 1	1	02356-01456	-	1
G008	00011100101001101001 0	1	01246-02356	-	1
G009	00011100101001101000 1	1	02356-0146	-	1
G010	00011100101001101001 1	1	02356-0146	-	1
G011	00011100101001000111 0	1	01346-0245	-	1
G012	00011100100001100111 0	1	01245-02346	-	1
G013	00011100101001000111 1	1	01346-0245	-	1
G014	00011100101001101011 1	1	01456-02356	-	1
G015	00011100110010101001 1	1	01246-01356	-	1
G016	00101100101001100000 0	1	02356-146	-	1
G017	00101100101001100000 1	1	12456-02356	-	1
G018	00101100111001100100 0	1	02356-1456	-	1
G019	00101100111001000100 1	1	1456-02356	-	1
G020	00101100110001100100 1	1	145-02356	-	1
G021	00101100110001000101 1	1	12456-0356	-	1
G022	00101100111001100100 1	1	1456-02356	-	1
G023	00101100111000101001 0	1	01356-1246	-	1
G024	00101100111000101001 1	1	01356-0246	-	1
G025	00110100111001100000 0	1	01346-1256	-	1
G026	00110100111001000100 0	1	01346-125	-	1
G027	00110100111001000000 1	1	12456-0346	-	1
G028	00110100110001100000 1	1	12456-0346	-	1
G029	00110100111001100100 0	1	1256-01346	-	1
G030	00110100111001100000 1	1	12456-01346	-	1
G031	00110100111001000100 1	1	1256-01346	-	1
G032	00110100111001100101 0	1	12456-01346	-	1
G033	00111100111001100001 0	1	01256-01346	-	1
G034	00111100111001000101 0	1	01346-01245	-	1
G035	00111100110001100101 0	1	02356-01456	-	1
G036	00111100111001100101 0	1	01256-01346	-	1
G037	00111100111001100100 1	1	01456-02356	-	1
G038	00111100111001100001 1	1	12456-03456	-	1
G039	00111100111000101000 1	1	01356-01246	-	1
G040	00111100111000101100 1	1	02356-01456	-	1
G041	00110100111001101100 0	1	12356-01346	-	1
G042	00110100111001001100 1	1	12356-01346	-	1
G043	00110100111001101100 1	1	12356-01346	-	1
G044	00110100111001101001 1	1	03456-12456	-	1
G045	00111000111000100111 0	1	12456-03456	-	1
G046	00111100111000100111 0	1	0236-01456	-	1
G047	00111100111001001101 1	1	01456-02356	-	1
G048	00111100111001100111 0	1	12456-03456	-	1
G049	00111100111001100110 1	1	02356-01456	-	1
G050	00111000111001100111 1	1	03456-12456	-	1
G051	00101100111011000001 0	1	01356-1246	-	1
G052	00101100111011000101 0	1	01346-01245	-	1
G053	00101100111010101001 1	1	01356-12456	-	1
G054	00101100111010100110 0	1	01356-12456	-	1
G055	00101100111010000110 1	1	1245-01356	-	1
G056	00101100110010100110 1	1	0356-12456	-	1
G057	00101100110010000111 1	1	12456-01356	-	1
G058	00101100111010100110 1	1	0356-12456	-	1
G059	00101100111010000111 1	1	12456-0356	-	1
G060	00101100110010100111 1	1	03456-12456	-	1
G061	00101000111010100111 1	1	12456-0356	-	1
G062	00101100111011000111 1	1	0356-12456	-	1
G063	00111100111010101111 0	1	03456-12456	-	1
G064	00111100111010101110 1	1	01356-01456-01246	-	1
G065	00111100111001110001 0	1	01346-01256	-	1
G066	00111100110001110100 1	1	02356-01345	-	1
G067	00111000111001010101 1	1	01256-01346	-	1
G068	00111100111001110101 0	1	01256-01346	-	1
G069	00111100111001110100 1	1	01256-01346	-	1
G070	00111100111001110001 1	1	01346-01256	-	1
G071	00111100110001110101 1	1	03456-12456	-	1
G072	00110101010011100100 0	1	02346-12356	-	1
G073	00111101011011000101 0	1	01346-01245	-	1
G074	00111101011011100100 1	1	01356-02456	-	1
G075	00111101001011101101 0	1	02456-01356	-	1
G076	00110101010011101101 0	1	02346-12356	-	1
G077	00110001011011101101 0	1	23456-03456-13456	-	1
G078	00110101011011101101 0	1	02346-12356	-	1
G079	00110101011011001101 1	1	01356-02456	-	1
G080	00111101011011101101 0	1	01356-02456	-	1
G081	00111101010011101101 1	1	01356-02456	-	1
G082	00111001011011101101 1	1	01356-02456	-	1
G083	00110101011011001111 1	1	01356-02456	-	1
G084	00111001011011110100 1	1	02345-12356	-	1
G085	00111101011011110100 1	1	02346-02356-01356	-	1
G086	00111101011011111101 0	1	23456-03456-13456	-	1
G087	01011101011011000101 1	1	02456-01356	-	1
G088	01011101111010001101 0	1	01246-01356	-	1
G089	01011101111010101101 0	1	01356-01246	-	1
G090	01011101111010101111 0	1	02456-13456	-	1

Name	Adjacency	#Aut	t.d.	Checks	#AutRem
G091	000110001000011001010	2	02456-02346-01246	4<6	1
G092	000101001000011010011	2	02356-01246	4<5	1
G093	000101001000011010111	2	23456-12456-02456	4<5	1
G094	001010001010011000000	2	12456-02346	5<6	1
G095	001011001010010001000	2	02356-01346	0<6	1
G096	001011001110001010000	2	01356-01456-01256	1<6	1
G097	001011001110011010010	2	01456-01256-01356	5<6	1
G098	001011001110011010011	2	01456-02356	5<6	1
G099	001101001100011001000	2	02356-01356-01346	0<6	1
G100	001100001110011001000	2	12356-01356-01456	1<6	1
G101	001101001100011001001	2	12456-01456-03456	0<5	1
G102	001101001110010000011	2	01346-12456	1<4	1
G103	001111001110010000011	2	01256-01456-03456	0<5	1
G104	001110001110011011010	2	01436-01456-12456	1<4	1
G105	001110001110011011011	2	01456-01356-12356	5<6	1
G106	001111001100011001111	2	02456-03456-01456	5<6	1
G107	001011001110101010000	2	01356-12456	1<6	1
G108	001010001100101001111	2	02346-03456-01345	5<6	1
G109	001111001100011100011	2	03456-02456-01456	0<6	1
G110	001111011100011011011	2	02356-01356-01346	5<6	1
G111	001101010110110001010	2	01346-12456	0<3	1
G112	001111010110110001001	2	01356-02456	0<5	1
G113	001111010110111010011	2	02456-01356	5<6	1
G114	001111010100111101001	2	01356-02356-2346	3<5	1
G115	001111010110111110011	2	03456-03456-23456	3<4	1
G116	010111011110101011000	2	01356-01246	0<1	1
G117	010111011110110011010	2	13456-02456	4<6	1
G118	011011011110111010011	2	12456-01256-01356	5<6	1
G119	011101011110111011000	2	01236-01234-12345	0<3	1
G120	011111011110111011010	2	01246-01236-01235	1<5	1
G121	001100001100011001000	14	01346-01236-12356	3<4	7
G122	011110011110111011010	14	01246-01236-01235	1<4	7

Below is the data for the graphs in which cycles of length 7 occurs odd number of times and didn't occur in the above graphs.

Name	Adjacency	#Aut	t.d.	Checks	#AutRem
G01	001101001100011001010	1	0346-12456	-	1
G02	001101001110011001001	1	0346-12456	-	1
G03	001101001110011010010	1	02356-01456	-	1
G04	001111001110011001111	1	01456-02356	-	1
G05	001011001110110001001	1	01356-1245	-	1
G06	001011001110110001011	1	0356-12456	-	1
G07	001011001110101010010	1	01356-1246	-	1
G08	001010001110101001110	1	12456-01356	-	1
G09	001010001110101001101	1	12456-0356	-	1
G10	001011001110101001110	1	12456-0356	-	1
G11	001011001110101001111	1	12456-0356	-	1
G12	001101001110110010011	1	1245-03456	-	1
G13	001111001110101011111	1	12456-03456	-	1
G14	001110001110011100011	1	12456-03456	-	1
G15	001110001100011101011	1	03456-12456	-	1
G16	001111001110011101011	1	01256-01346	-	1
G17	001111010110111001000	1	02456-01356	-	1
G18	001111010010111011001	1	02456-01356	-	1
G19	001111010110111011011	1	02456-01356	-	1
G20	001111010110101011110	1	01356-02456	-	1
G21	001111011010011111101	1	02356-03456-1346	-	1
G22	001110010110110101011	1	12456-01456-01346	-	1
G23	001111010110110101011	1	01356-03456-2345	-	1
G24	001111010110111111011	1	02346-02356-1356	-	1
G25	010111011010110010011	1	02456-13456	-	1
G26	011111011110111010011	1	01345-01456-02456	-	1
G27	001100001110011001010	2	12456-01346	1<4	1
G28	001101001110011001011	2	12456-0346	1<4	1
G29	001111001110011001000	2	01346-01256	0<6	1
G30	001110001110011001010	2	03465-12456	0<5	1
G31	001111001110011001011	2	01456-02356	0<5	1
G32	001101001110011010000	2	12356-01346	1<6	1
G33	001011001100101010010	2	01356-1246	0<6	1
G34	001011001100101001110	2	0145-02346	4<6	1
G35	001111010100111001001	2	12356-02356-0246	2<3	1
G36	001111010100101011111	2	01356-02456	5<6	1
G37	001111010110111101000	2	1356-02356-02346	2<5	1
G38	010111010110101010001	2	1356-02456	0<5	1
G39	010111011010110011010	2	02345-01346	0<6	1
G40	010111011110101011111	2	02456-13456	4<5	1
G41	011011011110110011011	2	02356-23456-13456	2<4	1
G42	011111011110111011011	2	01356-01256-01246	0<5	1