Hard

Table: Tasks

```
+----------------+---------+
| Column Name    | Type    |
+----------------+---------+
| task_id        | int     |
| subtasks_count | int     |
+----------------+---------+
```

task_id is the column with unique values for this table.
Each row in this table indicates that task_id was divided into subtasks_count subtasks labeled from 1 to subtasks_count.
It is guaranteed that 2 <= subtasks_count <= 20.

Table: Executed

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| task_id       | int     |
| subtask_id    | int     |
+---------------+---------+
```

(task_id, subtask_id) is the combination of columns with unique values for this table.
Each row in this table indicates that for the task task_id, the subtask with ID subtask_id was executed successfully.
It is **guaranteed** that subtask_id <= subtasks_count for each task_id.

Write a solution to report the IDs of the missing subtasks for each task_id.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

Input:
Tasks table:

```
+---------+----------------+
| task_id | subtasks_count |
+---------+----------------+
| 1       | 3              |
| 2       | 2              |
| 3       | 4              |
+---------+----------------+
```

Executed table:

```
+---------+------------+
| task_id | subtask_id |
+---------+------------+
| 1       | 2          |
| 3       | 1          |
| 3       | 2          |
| 3       | 3          |
| 3       | 4          |
+---------+------------+
```

Output:

```
+---------+------------+
| task_id | subtask_id |
+---------+------------+
| 1       | 1          |
| 1       | 3          |
| 2       | 1          |
| 2       | 2          |
+---------+------------+
```

Explanation:

Task 1 was divided into 3 subtasks (1, 2, 3). Only subtask 2 was executed successfully, so we include (1, 1) and (1, 3) in the answer.

Task 2 was divided into 2 subtasks (1, 2). No subtask was executed successfully, so we include (2, 1) and (2, 2) in the answer.

Task 3 was divided into 4 subtasks (1, 2, 3, 4). All of the subtasks were executed successfully.

```
# Write your MySQL query statement below
WITH recursive number AS(
    SELECT *
    FROM Tasks

    UNION

    SELECT task_id, subtasks_count  - 1
    FROM number
    WHERE subtasks_count  > 0
),
CTE AS (
        SELECT task_id,RANK () OVER (PARTITION BY task_id ORDER BY subtasks_count) AS
subtask_id
        FROM number
        WHERE subtasks_count!=0 )

SELECT *
FROM CTE
WHERE (task_id,subtask_id) NOT IN (SELECT *
                                   FROM Executed)
```