Problem 465: Optimal Account Balancing

Problem Information

Difficulty: Hard
Acceptance Rate: 50.15%
Paid Only: Yes
Tags: Array, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask

Problem Description

You are given an array of transactions `transactions` where `transactions[i] = [fromi, toi, amounti]` indicates that the person with `ID = fromi` gave `amounti $` to the person with `ID = toi`.

Return _the minimum number of transactions required to settle the debt_.

Example 1:

Input: transactions = [[0,1,10],[2,0,5]] **Output:** 2 **Explanation:** Person #0 gave person #1 $10. Person #2 gave person #0 $5. Two transactions are needed. One way to settle the debt is person #1 pays person #0 and #2 $5 each.

Example 2:

Input: transactions = [[0,1,10],[1,0,1],[1,2,5],[2,0,5]] **Output:** 1 **Explanation:** Person #0 gave person #1 $10. Person #1 gave person #0 $1. Person #1 gave person #2 $5. Person #2 gave person #0 $5. Therefore, person #1 only need to give person #0 $4, and all debt is settled.

Constraints:

* `1 <= transactions.length <= 8` * `transactions[i].length == 3` * `0 <= fromi, toi < 12` * `fromi != toi` * `1 <= amounti <= 100`

Code Snippets

C++:

```
class Solution {
public:
int minTransfers(vector<vector<int>>& transactions) {

}
};
```

Java:

```
class Solution {
public int minTransfers(int[][] transactions) {

}
}
```

Python3:

```
class Solution:
def minTransfers(self, transactions: List[List[int]]) -> int:
```