Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2023), B.Sc. in CSE (Day)

Cash Flow Minimizer

Course Title: Algorithm Lab
Course Code: CSE206
Section:221 D8

Students Details

Name	ID
Md. Mahim Hossain	221902082

Submission Date: 25 Dec 2023
Course Teacher's Name: Md. Abu Rumman Refat

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The Cash Flow Minimizer system is designed to optimize and minimize the number of transactions among multiple banks with varying payment modes. The system introduces an intermediary world bank to facilitate transactions between banksthat lack a common payment mode. The goal is to efficiently balance cash flows across different banks and payment types.

1.2 Motivation

The motivation behind the Cash Flow Minimizer system lies in simplifying financial transactions among diverse banks with distinct payment modes. By minimizing the number of transactions required for balancing cash flow, the system aims to enhance efficiency in global financial interactions, reduce transaction costs, and streamline the overall banking process.

1.3 Problem Definition

1.3.1 Problem Statement

The intricacy stems from managing a varied array of banks, each with multiple payment modes. Navigating the interplay of diverse payment types and the necessity for a methodical approach to minimize cash flow complexity pose a challenging engineering problem. This complexity underscores the need for a sophisticated solution that addresses the intricacies of financial transactions within this diverse banking landscape.

1.3.2 Complex Engineering Problem

This project(CFM) tackles a complex engineering problem by developing an algorithm that optimizes cash flow among banks with diverse payment methods. The solution streamlines financial transactions, minimizing complexities in inter-bank dealings. By offering a systematic approach, the project contributes to enhancing efficiency and effectiveness in managing varied payment modalities within a network of banks. The result is a scalable solution that significantly improves the overall functionality of financial transactions.

1.4 Design Goals

- Minimize the number of transactions required to balance cash flow.
- Handle diverse payment modes among banks.
- Develop an efficient and scalable algorithm for cash flow minimization.
- Ensure accurate and reliable results.

1.5 Application

The Cash Flow Minimizer system finds applications in international banking, where banks operate with various payment modes. It can be utilized to optimize transactions in a global financial network, reducing the complexity and costs associated with cross-border transactions.

Table 1.1: A summary of the attributes touched by the provided Java project, along with an explanation of how to address them. Let's create a table for better clarity:

Name of the Project Attribute	Explanation of How to Address
Bank Class	The Bank class represents a bank with attributes such as name, netAmount, and types. To address, ensure proper encapsulation of attributes, use accessors, and mutators to manage data integrity.
”Pair” Class	The Pair class is a generic class for holding key-value pairs. To address, consider using standard Java classes like Map.Entry or AbstractMap.SimpleEntry instead of defining a custom Pair class.
”CashFlowMinimizer” Class	The main class orchestrating the cash flow minimization algorithm. To address, consider breaking down into smaller methods for readability and maintainability.
”getMinIndex” and ”getSimpleMaxIndex” Methods	These methods find the minimum and maximum net amounts respectively. To address, ensure code efficiency and readability. Consider early return statements for optimization.
”getMaxIndex” Method	Finds the maximum net amount with a common payment type. To address, ensure clarity and efficiency. Consider breaking down into smaller methods for readability.
”printAns” Method	Prints the results of cash flow minimization algorithm. To address, ensure proper formatting and consider breaking down into smaller methods for readability.
”minimizeCashFlow” Method	The main cash flow minimization algorithm. To address, consider modularizing and breaking down into smaller methods for readability and maintainability.
”main” Method	The entry point of the program, taking user input for banks and transactions. To address, consider validating and error handling for user input.

Chapter 2

Implementation of the Project

2.1 Introduction

The project employs a combination of greedy and iterative algorithms to iteratively identify and execute transactions that reduce the net amounts among banks.

2.2 Project Details

The Cash Flow Minimizer project optimizes international financial transactions among diverse banks by minimizing the number of transactions required to balance cash flows.

The project employs a greedy algorithm for cash flow minimization due to its effectiveness in making locally optimal choices at each step. In the context of settling debts among banks, the algorithm prioritizes transactions with immediate impact, aiming for an overall optimal solution. This approach streamlines the minimization process by efficiently reducing net amounts between banks in a step-by-step fashion.

Greedy Algorithm (Used in the original code):

- Time Complexity: O(V$\hat{2}$), where V is the number of banks.
- The greedy approach involves multiple iterations, and the complexity is influenced by the nested loops in the code.

Ford-Fulkerson Algorithm (as a potential alternative):

- Time Complexity: O(E * f), where E is the number of edges and f is the maximum flow in the network.
- The Ford-Fulkerson algorithm is typically applied to network flow problems and might require additional modifications for cash flow minimization.

2.2.1 Cash Flow Minimization

The primary focus is on minimizing the net amounts among banks through an iterative process of identifying pairs of banks and executing transactions.

You can fix the height, width, position, etc., of the figure accordingly.

2.3 Implementation

The implementation utilizes a combination of greedy and iterative algorithms to iteratively identify and execute transactions, minimizing net amounts among banks in the Cash Flow Minimizer project.

2.3.1 Algorithmic Approach

The implementation involves the use of three key algorithms: one for finding the bank with the minimum net amount, another for finding the bank with the maximum net amount, and a third for finding the bank with the maximum net amount and matching payment types.

2.4 Algorithms

The project employs a greedy approach to iteratively minimize cash flow by selecting banks with the minimum and maximum net amounts and conducting transactions.

Procedure for Cash Flow Minimization System:

Welcome and System Introduction

- Display a welcome message to the user explaining the purpose of the Cash Flow Minimization System.

User Input for Bank Details

- Prompt the user to input the number of participating banks.
- For each bank:
  - Request the bank's name.
  - Ask for the number of payment modes it supports.
  - Gather information about each payment mode the bank accepts.

User Input for Transaction Details

- Prompt the user to input the number of transactions to consider.
- For each transaction:
  - Obtain the originating bank's name.
  - Identify the receiving bank's name.
  - Collect the transaction amount.

Initialization and Data Preparation

- Create instances of the Bank class for each bank, storing relevant information (name, net amount, payment types).
- Initialize a 2D array (graph) to represent the transactions between banks.

Cash Flow Calculation

- Call the `minimizeCashFlow` method:
  - Calculate the net amount for each bank based on incoming and outgoing transactions.
  - Determine the minimum cash flow transactions among the banks.

Print Minimum Cash Flow Transactions

- Display the results using the `printAns` method:
  - Print the transactions that minimize the overall cash flow among the banks.

End of Procedure

- Conclude the system interaction.

Procedure for Cash Flow Minimization System:

'getMinIndex' Method

- Find the index of the bank with the minimum net amount.

'getSimpleMaxIndex' Method

- Find the index of the bank with the maximum net amount.

'getMaxIndex' Method

- Find the index and common payment type for the bank with the maximum net amount.

'printAns' Method

- Print the transactions for minimum cash flow based on the calculated results.

'minimizeCashFlow' Method

- Calculate and print the minimum cash flow transactions among multiple banks:
  - Initialize necessary data structures.
  - Iterate through banks to determine minimum and maximum net amounts.
  - Identify common payment types between banks.
  - Update net amounts based on transactions.
  - Print the optimized transactions.

'Main' Method

- The main entry point of the program:
  - Display system introduction and purpose.
  - Accept user input for bank and transaction details.
  - Call the `minimizeCashFlow` method to perform cash flow minimization.

Implementation:

- Welcome Message

```
    System.out.println("\n\t\t\t\t********************* Welcome to CAS
System.out.println("This system minimizes the number of transactions a
intermediary between banks that have no common mode of payment.\n");
```

- Input Collection

```
   System.out.println("Enter the number of banks participating in the
int numBanks = scanner.nextInt();
```

- Bank Details Input

```
System.out.println("Enter the number of transactions."); int numTransa

int[][] graph = new int[numBanks][numBanks];

System.out.println("Enter the details of the transactions as stated:")
System.out.println("Bank name from which the transaction originates, b

for (int i = 0; i < numTransactions; i++) { String from = scanner.next
String to = scanner.next();
int amount = scanner.nextInt();

Integer fromIndex = indexOf.get(from); Integer toIndex = indexOf.get(t
null) { int fromIdx = fromIndex.intValue();
int toIdx = toIndex.intValue();
graph[fromIdx][toIdx] = amount;
} else {
System.out.println("Invalid bank names: " + from + ", " + to);
}
}
```

- Minimize Cash Flow Calculation

```
minimizeCashFlow(numBanks, input, indexOf, numTransactions, graph, max
```

- Print Result

```
scanner.close();
```

- Result Printing Method

```
public static void printAns(List<List<Pair<Integer, String>>> ansGraph
```

Chapter 3

Performance Evaluation

3.1 Simulation Environment

3.1.1 Input Simulation

The system is tested with simulated input scenarios representing different banks, payment modes, and transaction details.

3.1.2 Algorithm Performance Testing

The algorithms are evaluated in terms of their efficiency in minimizing cash flow and handling diverse banking scenarios.

3.1.3 Algorithm Performance Testing

The algorithms are evaluated in terms of their efficiency in minimizing cash flow and handling diverse banking scenarios.

3.2 Results Testing

3.2.1 Transaction Results

The output includes details of transactions performed to minimize cash flow among banks.

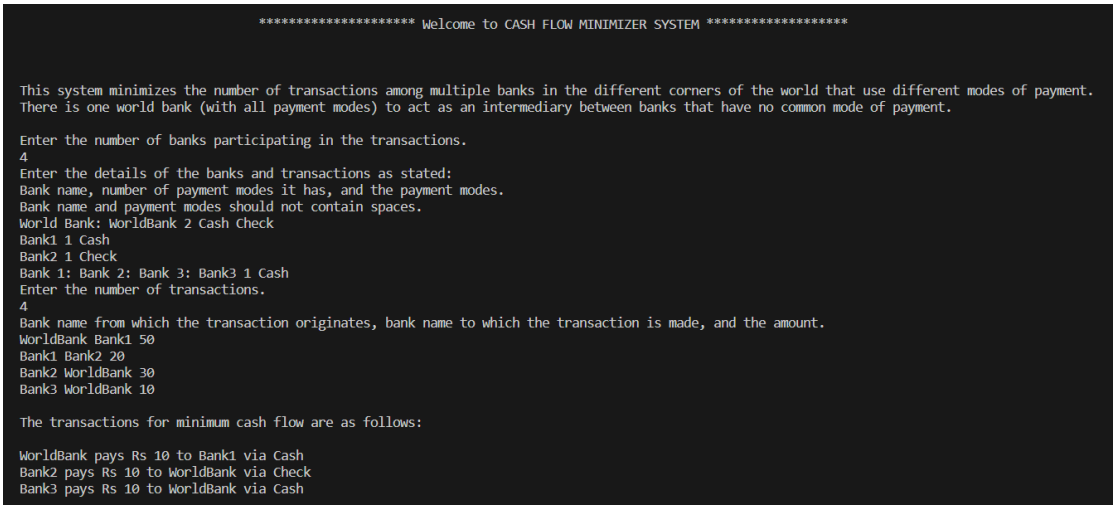

Figure 3.1: Enter Caption

3.2.2 Algorithm Efficiency

The efficiency of the algorithms is assessed based on their ability to handle varying input sizes and complexities.

3.2.3 Performance Metrics

Key performance metrics, such as execution time and transaction accuracy, are analyzed.

3.3 Results Overall Discussion

3.3.1 Complex Engineering Problem Discussion

The system's efficacy in tackling the intricate engineering challenge of minimizing cash flow is examined, taking into account its real-world applications and practical implications.

Chapter 4

Conclusion

4.1 Discussion

The Cash Flow Minimizer system demonstrates an effective approach to minimizing cash flow imbalances among banks. The discussed algorithms and implementation strategies contribute to efficient cross-border transactions.

4.2 Limitations

- The system assumes non-negative transaction amounts.
- Limited handling of edge cases involving specific types of payment modes.

4.3 Scope of Future Work

Future work could focus on:

- Handling negative transaction amounts.
- Enhancing the system to consider additional factors influencing cash flow.

In conclusion, the Cash Flow Minimizer system offers a valuable solution for optimizing financial transactions among diverse banks, providing a foundation for further research and development in the field.

4.3.1 References

- Google.
- YouTube.
- GitHub.