**DEPARTMENT OF**

**SCHOOL OF COMPUTING**

**College of Engineering and Technology**

**SRM Institute of Science and Technology**

MINI PROJECT REPORT

ODD Semester, 2023-2024

Lab code & Sub Name : 21CSS201J & Data Structures and Algorithms

Year & Semester : II & III

Project Title : **Cash Flow Minimizer**

Lab Supervisor **:**

Team Members : 1. JEYASHAATHVEE (RA2211003011660)

2. YOGESHWARAN (RA2211003011672)

3. ANISH PRANAV (RA2211003011624)

|  |  |  |
| --- | --- | --- |
| **Particulars** | **Max. Marks** | **Marks Obtained** |
| **Name:** |
| **Register No :** |
| Problem selection | 2 |  |
| Demo | 1.5 |  |
| Project Reports | 1.5 |  |
| Subject knowledge | 2 |  |
| **Total** | **7** |  |

**Date :**

**Staff Name :**

**Signature :**

**Cash Flow Minimizer**

**OBJECTIVE:**

The objective of a Cash Flow Minimizer is to efficiently manage and optimize the allocation of cash flows within a financial system to minimize the total amount of cash held in the system while ensuring that all financial obligations are met.

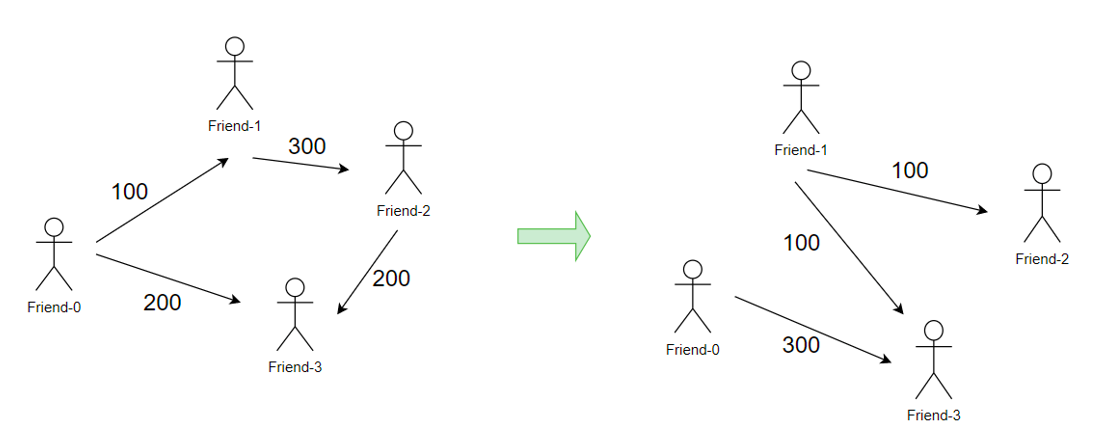
**PROBLEM DEFINITION:**

In a financial system with multiple entities engaged in various financial transactions, the Cash Flow Minimizer aims to optimize the allocation of cash flows to minimize the total amount of cash held in the system while ensuring that all financial obligations are met, including timely repayments of loans, payment of bills, and other commitments. The objective is to design an efficient data structure and associated algorithms that can handle a diverse range of financial instruments, uncertainties in cash flow patterns, and provide real-time updates to achieve the most efficient use of cash within the system.

**PROBLEM EXPLANATION :**

In this application, you can find minimum cash flow between n people. Insert data in the table where cell represent the value person ‘i’ has to pay to person ‘j’. On computing the result, you will get answer with minimum cash flow between persons. This uses Graph and Greedy method. Time complexity is O(n^2). Using min heap and max heap, the time complexity can be reduced to O(nlogn).

**EXAMPLE :**



**DATA STRUCTURE USED:**

The approach we will be using for minimizing cash flow is the Greedy Algorithm.

The greedy approach is used to build the solution in pieces, and this is what we want to minimize the cash flow. At every step, we will settle all the amounts of one person and recur for the remaining n-1 persons.

Calculate the net amount for every person, which can be calculated by subtracting all the debts, i.e., the amount to be paid from all credit, i.e., the amount to be paid to him. After this, we will find two persons with the maximum and the minimum net amounts. The person with a minimum of two is our first person to be settled and removed from the list.

**APPROACH/METHODOLOGY/PROGRAMS:**

#include <stdio.h>

// Function to find the maximum and minimum values in the given array

void findMaxMin(int arr[], int n, int\* maxIndex, int\* minIndex) {

\*maxIndex = 0;

\*minIndex = 0;

for (int i = 1; i < n; i++) {

if (arr[i] > arr[\*maxIndex]) {

\*maxIndex = i;

}

if (arr[i] < arr[\*minIndex]) {

\*minIndex = i;

}

}

}

// Function to minimize cash flow among a group of people

void minimizeCashFlow(int amount[], int n) {

int maxCreditIndex, maxDebitIndex;

findMaxMin(amount, n, &maxCreditIndex, &maxDebitIndex);

if (amount[maxCreditIndex] == 0 && amount[maxDebitIndex] == 0) {

return; // All balances settled

}

int minBalance = (amount[maxCreditIndex] < -amount[maxDebitIndex]) ? amount[maxCreditIndex] : -amount[maxDebitIndex];

// Update the balances

amount[maxCreditIndex] -= minBalance;

amount[maxDebitIndex] += minBalance;

printf("Person %d pays %d to Person %d\n", maxDebitIndex, minBalance, maxCreditIndex);

minimizeCashFlow(amount, n); // Recursively minimize the remaining balances

}

int main() {

int n; // Number of people

printf("Enter the number of people: ");

scanf("%d", &n);

int amount[n]; // Balances for each person

printf("Enter the balances for each person:\n");

for (int i = 0; i < n; i++) {

scanf("%d", &amount[i]);

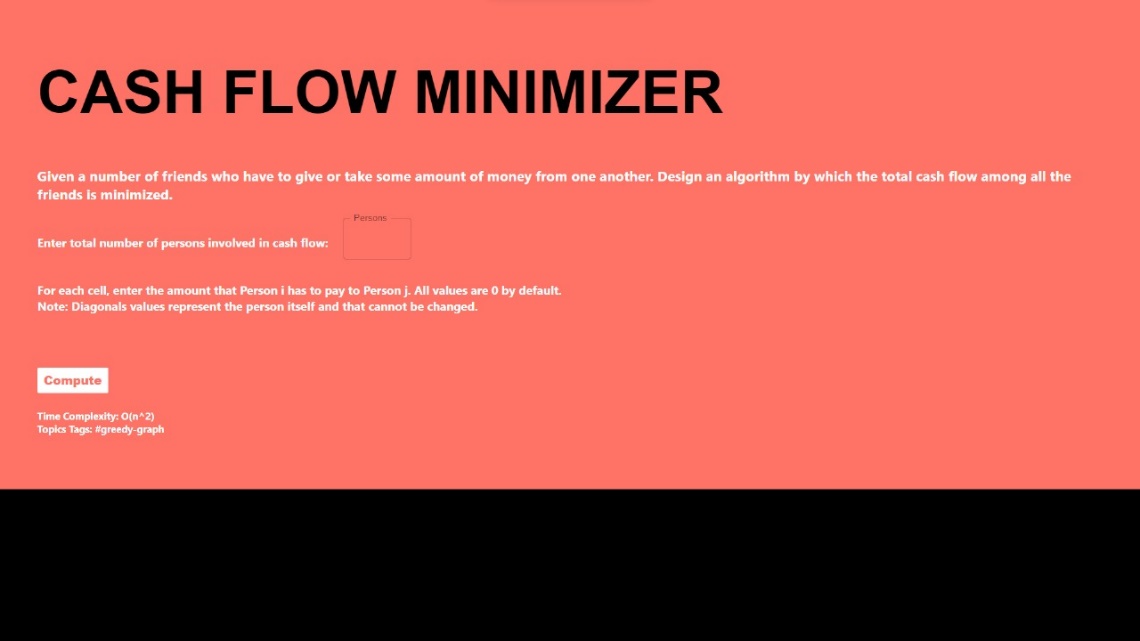
}

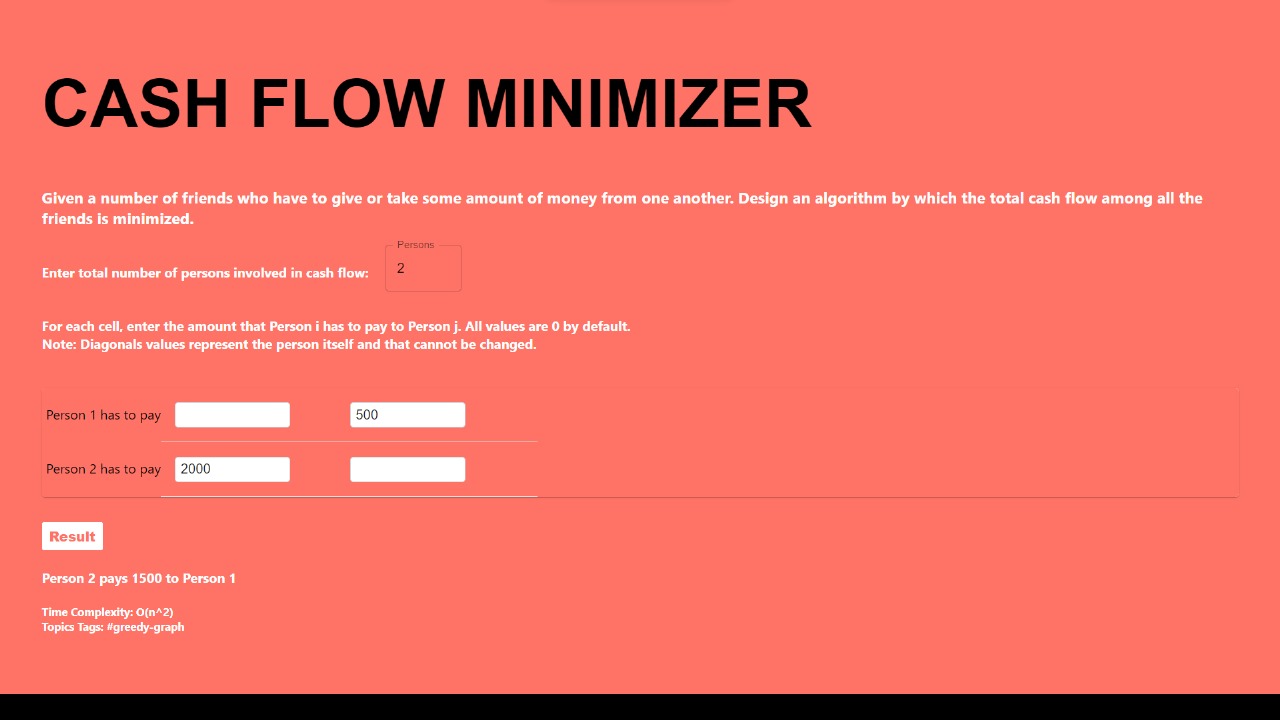
minimizeCashFlow(amount, n);

return 0;

}

**OUTPUT:**

****

****

**CONCLUSION:**

By implementing strategies to optimize cash flow, control costs, and improve financial stability, we have enhanced the project's ability to meet its financial obligations and ensure timely execution. This project's outcomes not only minimize cash-related challenges but also contribute to the overall success and sustainability of future projects.