Final Project -- Splitwise Algorithm Design

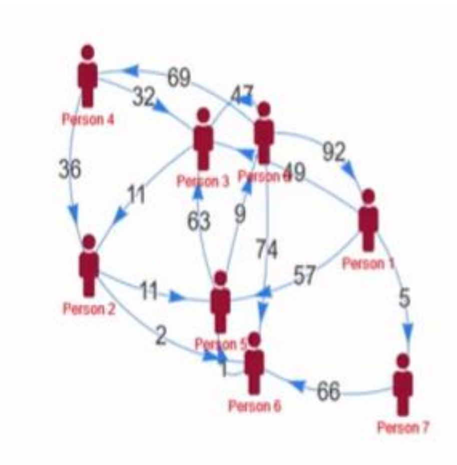
1. Description

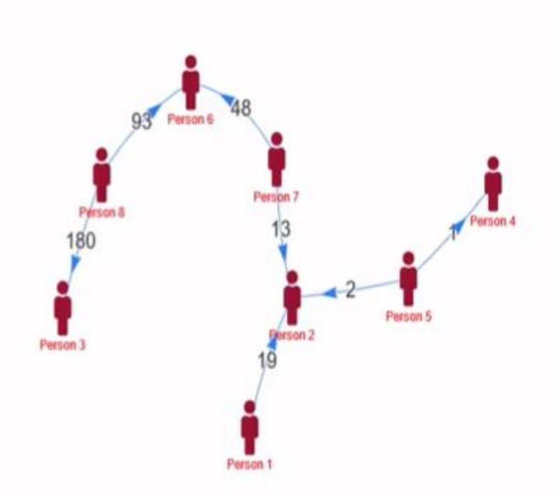
In life, we often encounter situations where we borrow money from friends because we don't have any change. Or when you go out with a group of friends, one friend pays first and then the others split the money equally. When there are a lot of people, it will cause a complicated cash flow. Therefore, I would like to **design an algorithm to simplify the cash flow.**

2. Skills to be use

- Graph & RB tree & B tree
- Splitwise algorithm to remove unnecessary edges(cash flow) of graph.

3. Goal





4. Algorithm

1. Calculate the amount owed by each person.
2. Debit = min(amount), credit = max(amount), settlement_amount = min{|debit|, credit}
3. Settlement amount paid by the debtor to the creditor.
4. Go back to step 2 until the amount owed is paid in full.

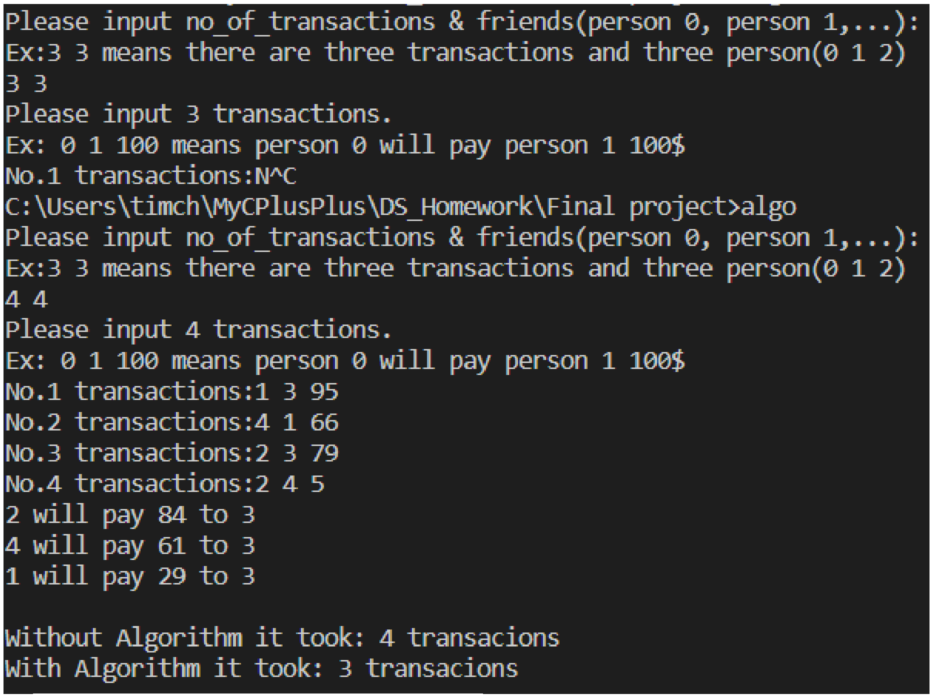
5. Time complexity

1. Input transactions and add in map(Red-Black tree). **O(n) * O(logn)** (n:number of transactions)
2. Add amount to multiset(B tree). **O(n) * O(logn)**
3. Settlement amount paid by the debtor to the creditor. **O(1)**
4. Go back to step 3 until the amount owed is paid in full. **O(n)**

Time complexity = O(nlogn) + O(nlogn) + O(1) * O(n) = **O(nlogn)**

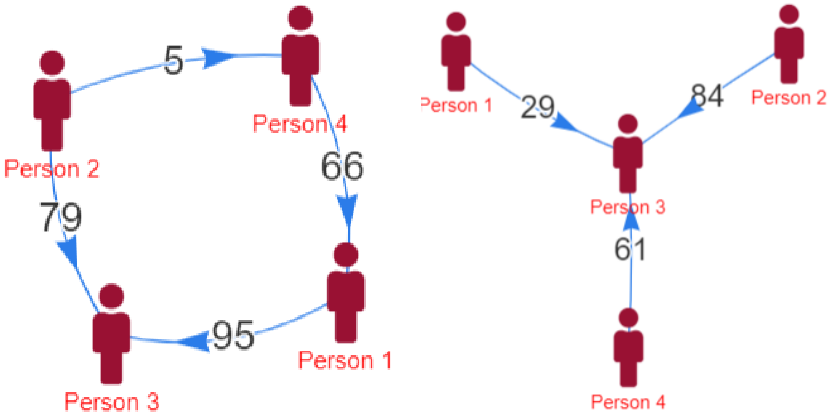
6. Result

Ex1:



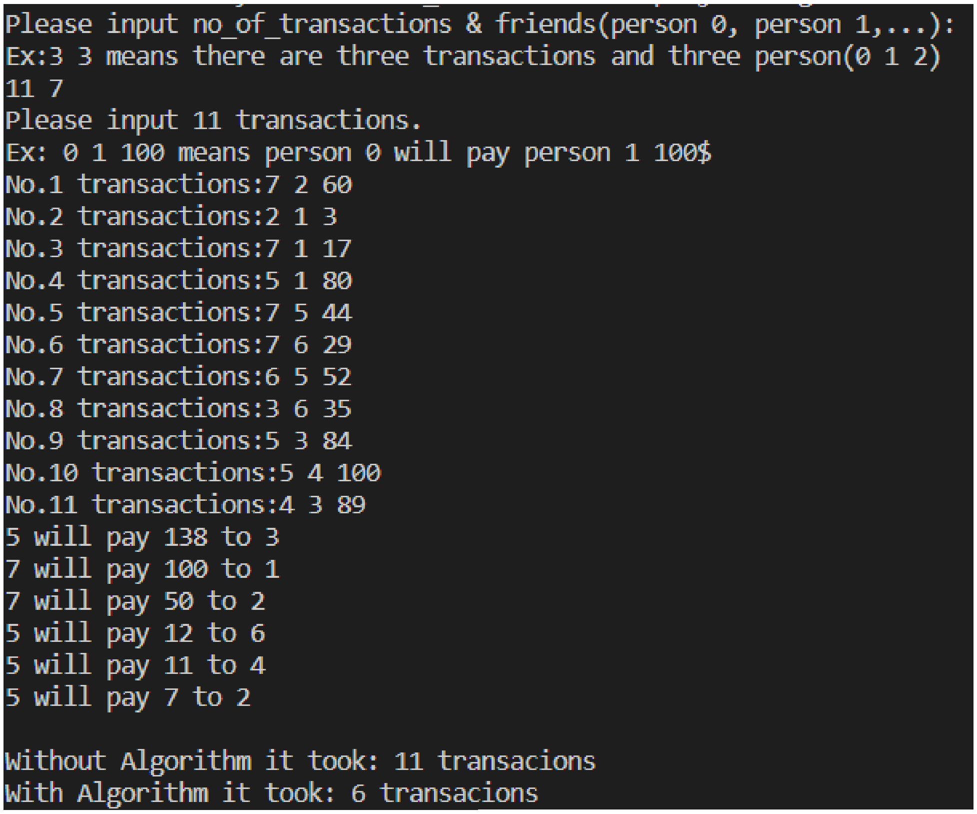
```
Please input no_of_transactions & friends(person 0, person 1,...):
Ex:3 3 means there are three transactions and three person(0 1 2)
3 3
Please input 3 transactions.
Ex: 0 1 100 means person 0 will pay person 1 100$
No.1 transactions:N^C
C:\Users\timch\MyCPlusPlus\DS_Homework\Final project>algo
Please input no_of_transactions & friends(person 0, person 1,...):
Ex:3 3 means there are three transactions and three person(0 1 2)
4 4
Please input 4 transactions.
Ex: 0 1 100 means person 0 will pay person 1 100$
No.1 transactions:1 3 95
No.2 transactions:4 1 66
No.3 transactions:2 3 79
No.4 transactions:2 4 5
2 will pay 84 to 3
4 will pay 61 to 3
1 will pay 29 to 3

Without Algorithm it took: 4 transacions
With Algorithm it took: 3 transacions
```



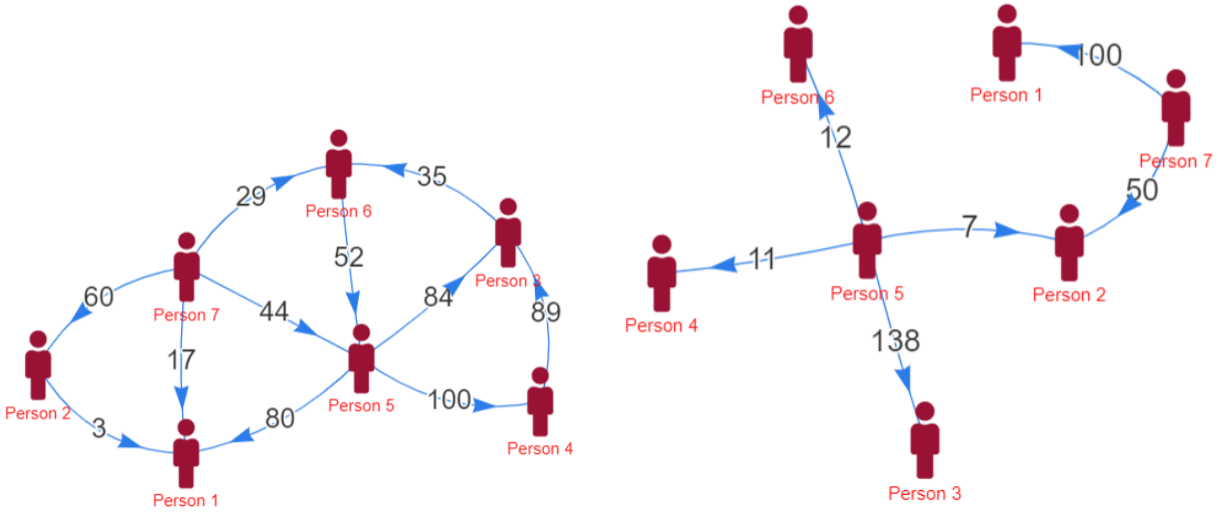
P.S. The above two diagrams are schematics, not the result of running the code

Ex2:



```
Please input no_of_transactions & friends(person 0, person 1,...):
Ex:3 3 means there are three transactions and three person(0 1 2)
11 7
Please input 11 transactions.
Ex: 0 1 100 means person 0 will pay person 1 100$
No.1 transactions:7 2 60
No.2 transactions:2 1 3
No.3 transactions:7 1 17
No.4 transactions:5 1 80
No.5 transactions:7 5 44
No.6 transactions:7 6 29
No.7 transactions:6 5 52
No.8 transactions:3 6 35
No.9 transactions:5 3 84
No.10 transactions:5 4 100
No.11 transactions:4 3 89
5 will pay 138 to 3
7 will pay 100 to 1
7 will pay 50 to 2
5 will pay 12 to 6
5 will pay 11 to 4
5 will pay 7 to 2

Without Algorithm it took: 11 transacions
With Algorithm it took: 6 transacions
```



P.S. The above two diagrams are schematics, not the result of running the code

7. Team

- Team Size:1
- Student I.D.:108501026
- Name:陳宏釜