

REPORT

BRIEF OVERVIEW:

This project aims to successfully implement a currency conversion system, where one can add and remove trade banks such as SBI, JP Morgan & Chase etc. and add various currencies such as the Rupee, Dollar, Pound etc. to these banks along with their conversion rates.

The project then aims to use Dijkstra's Algorithm to find the best path towards converting one currency into another.

DATA STRUCTURES USED IN THE PROJECT:

Linked lists

Array of linked lists to store the graph of currency nodes

Array of vertex ids of the currencies

For adjacency list:

Graph List - number of vertices, vertex array pointer

Graph Node (Vertex) - ID, weight, Next pointer

For currency linked list:

Currency Node - name, vertex ID, Next pointer

For vertex ID:

Array of 1000 vertices initialized to 0. //1 indicates that the vertexID is used up

For Trade Bank linked list:

Trade Bank Node - name, graph list pointer, currency list pointer, array of available vertices (boolean - 0 or 1), Next pointer

For Dijkstra's algorithm to determine min cost of conversion:

Priority Queue:

Heap Node - key, value

MinHeap - array of Heap Nodes, capacity of heap, current number of elements in heap

For detecting cycles in currency graph:

Stack - value, Next pointer.

DIVISION OF WORK:

- 1) Rudransh Pratap Singh: Dijkstra's Algorithm (including priority queue implementation), Stack implementation for SCC, testing.
- 2) Vikram Rao: Wrote RemoveCurrencyExchange function, PrintTradeBankList function, RemoveTradeBank function, README, testing.
- 3) Parth Maradia : AddTradeBank , AddCurrencyExchange, UserInterface.h, mainInputprogram.c , Testing , Commenting and debugging.
- 4) Anish Mathur: Wrote AddCurrencyExchange function, RemoveTradeBank function, PrintTradeBankList function, Commenting, testing.
- 5) Eswara Rohan: main.c, README, testing.

ALGORITHMS USED AND THEIR TIME COMPLEXITY

1. Dijkstra's Algorithm : $O(E \log E)$ //as we are using the lazy implementation
2. Overall Time Complexity : $O(n(E \log E))$ //n is number of trade banks
3. Tarjan SCC : $O(V+E)$ //V would be number of currencies in the trade bank and E number of edges