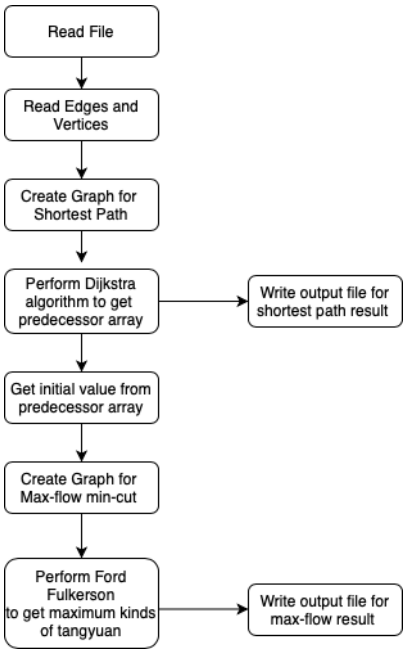
演算法Lab3

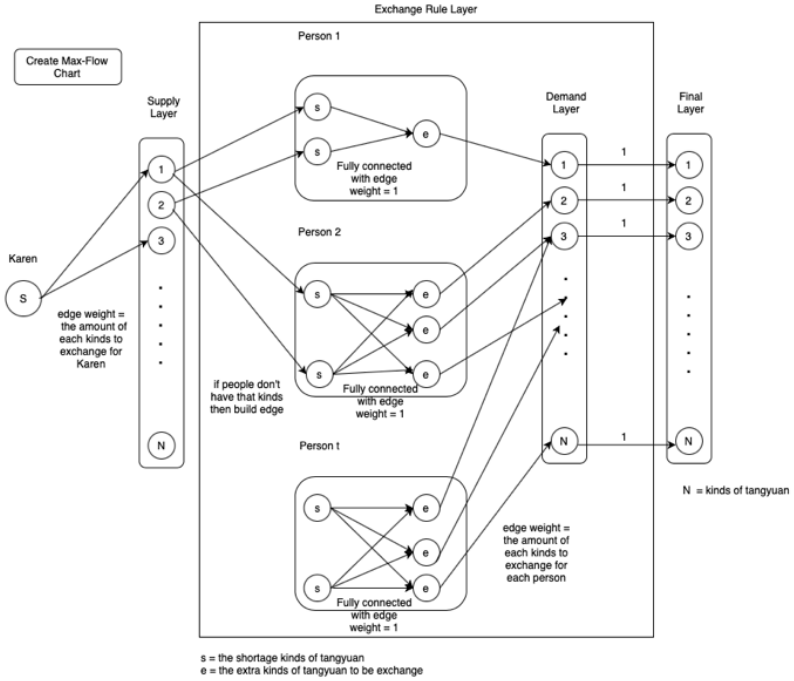
- 電機4C 洪愷尹 0710851

- Algorithm includes two part:

1. Shortest Path Problem
    - Perform pure Dijkstra algorithm to find shortest path to each supermarket and party.
    - Backtrace the shortest path from t to s and find what kinds of tangyuan and how many of them Karen bought.
    - Time Complexity : Since Dijkstra is implemented by the data structure of Min-Priority Queue -> O(V^2+E)
    - V represents home, party, supermarkets.
    - E represents the path between two locations.
2. Max-flow Min-cut Problem
    - Use the shortest path result to be the initial value Karen has.
    - Perform Ford Fulkerson algorithm to exchange the maximum kinds of Tangyuans.
    - Time Complexity : Use Edmonds-Karp type's algorithm, and use BFS to find augmenting paths from residual network -> O(VE^2)
    - V & E show in flow chart.
    - |V| = the number of people * the number of kinds of tangyuan + 2
    - |E| => max|E| = O(V^2)

- Flow Chart



- 
Exchange Rule Layer
Person 1
Create Max-Flow Chart
Supply Layer
Demand Layer
Final Layer
Karen
S
edge weight = the amount of each kinds to exchange for Karen
Fully connected with edge weight = 1
Person 2
if people don't have that kinds then build edge
Fully connected with edge weight = 1
Person t
Fully connected with edge weight = 1
edge weight = the amount of each kinds to exchange for each person
N = kinds of tangyuan
s = the shortage kinds of tangyuan
e = the extra kinds of tangyuan to be exchange