CSCE 2110.501 (13359)
Zion Welsh
Alan Wu
Caleb Layfield

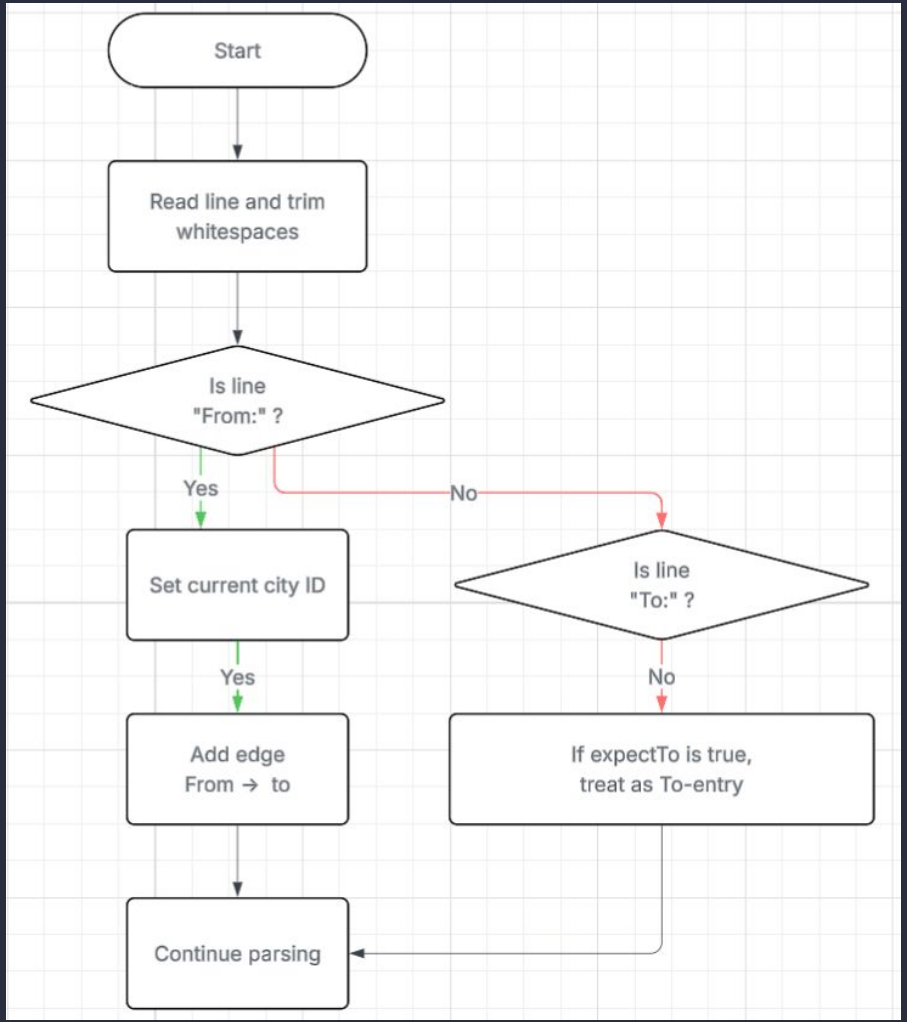
Route Search System Design & Implementation Presentation

Project overview and Key Designs

- Purpose: Graph based flight route system
- Cities = nodes, flights = edges
- Supports BFS, DFS, multistop routing
- Our Chosen design:
  - Adjacency list graph
  - Hash map city indexing
  - State based parsing

Flowchart: Input Parsing

- Handles the flexible From/To format
- Uses trimming + state switching
- Builds graph during parsing



Flow charts were made via: lucidchart.com

Data Structures

01. **vector<string> names**

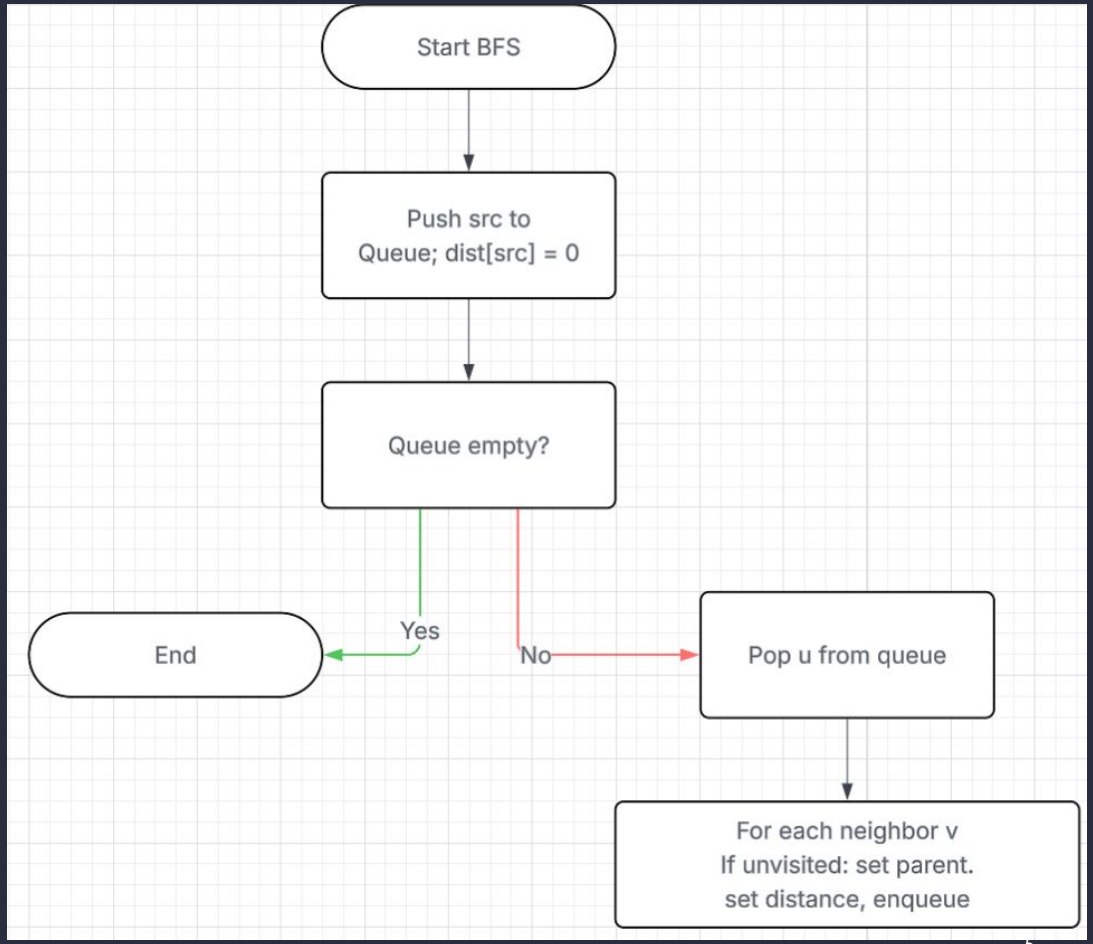
02. **unordered_map<string,int> idx**

03. **Vector<vector<int>> adj**

04. **Country grouping**

Algorithms and flowchart

- BFS ⇨ shortest routes (this is the core algorithm)
- DFS ⇨ reachability tour
- Multi BFS ⇨ routing through B and C
- Q4 ⇨ best meeting city using sum of BFS distances



Flow charts were made via: lucidchart.com

RunTime, Alternatives, and Optimization

RunTime:

Everything O(V+E)

Alternatives:

- Adjacency matrix = Too much memory
- Dijkstra/A = not needed (unweighted)
- Floyd Warshall = O(V^3), too slow

Optimizations:

- Parallel BFS
- BFS memoization
- Graph compression

Scalability and Conclusion

Adjacency list scales well

Hash map remains Efficient

For millions of cities -> compression/distributed graph

Final conclusions