需求分析和概要设计

需求分析

a. 问题

设计一个交通咨询系统，能够让旅客咨询从任一个城市到另一个城市之间的最低费用或最少时间等问题。设计分为三个部分：

1.建立交通网络图的存储结构。

2.解决单源最短路径问题。

3.实现两个城市之间的最短路径问题。

b. 分析

1.城市信息管理：

需要具备添加、修改、删除城市信息（城市名）的功能。

城市信息的存储结构应支持快速查询和更新。

2.交通信息管理：

需要具备添加、修改、删除航班和列车时刻表（航班和列车班次）的功能。

交通信息的存储结构应支持不同交通工具的区分和时刻表的管理。

3.最优决策：

提供最快到达或最省钱到达的咨询服务。

考虑旅途中的中转站等候时间（飞机至少2小时，火车至少1小时）。

使用Dijkstra算法解决单源最短路径问题，并根据不同的优化目标（时间或费用）进行计算。

4.用户交互：

提供人机交互界面，用户可以选择最优决策原则和交通工具，输入起始站、终点站、出发时间，输出相关信息并能够处理异常输入。

分解

1. 数据存储结构

城市信息：

使用链表存储城市信息。

支持城市信息的添加、修改和删除操作。

交通信息：

使用邻接表存储交通信息，邻接表的每个节点包含到达城市、费用和时间等信息。

支持不同交通工具的区分和时刻表的管理。

2. 城市信息管理模块

添加城市：

接受城市名称，添加到链表。

修改城市：

接受新的城市名称，更新链表。

删除城市：

接受城市，从链表中删除对应城市。

3. 交通信息管理模块

添加交通信息：

接受交通工具类型、起始城市、终点城市、费用和时间，添加到邻接表中。

修改交通信息：

接受交通信息的索引和新的交通信息，更新邻接表中的对应条目。

删除交通信息：

接受交通信息的索引，从邻接表中删除对应条目。

4. 最优决策模块

Dijkstra算法：

实现Dijkstra算法，用于计算从起始城市到其他城市的最短路径。

支持根据不同的优化目标（时间或费用）进行计算。

最优路径查询：

接受起始城市、终点城市和优化目标，调用Dijkstra算法进行计算，返回最优路径和相关信息（时间或费用）。

5. 用户交互模块

用户界面：

提供菜单选项，用户可以选择添加、修改、删除城市信息和交通信息，以及查询最优路径。

接受用户输入，调用相应的功能模块进行操作，并输出结果。

异常处理：

处理用户输入的异常情况，如无效的城市信息、索引超出范围等，提示用户重新输入。

概要设计

1. 框图

数据存储

用户界面

城市信息管理

交通信息管理

最优决策

Dijkstra算法

最优路径查询

1. 各模块分解设计思路

1.数据存储模块：

城市信息存储：

使用链表存储城市信息，支持快速查询、更新和展示。

交通信息存储：

使用邻接表存储交通信息，支持不同交通工具的区分和时刻表的管理。

2.用户界面模块：

提供菜单选项，用户可以选择不同的功能。

接受用户输入，调用相应的功能模块进行操作。

处理异常输入，提示用户重新输入。

3.城市信息管理模块：

添加城市：

接受城市名称，验证输入的有效性，例如是否重复，添加到链表中。

修改城市：

接受新的城市名称，验证输入的有效性，更新链表中的对应城市。

删除城市：

接受城市名称，验证输入的有效性，从链表中删除对应城市。

4.交通信息管理模块：

添加交通信息：

接受交通工具类型、起始城市、终点城市、里程、费用和时间，验证输入的有效性，添加到邻接表中。

修改交通信息：

接受交通信息的索引和新的交通信息，验证输入的有效性，更新邻接表中的对应条目。

删除交通信息：

接受交通信息的索引，验证输入的有效性，从邻接表中删除对应条目。

5.最优决策模块：

Dijkstra算法：

实现Dijkstra算法，用于计算从起始城市到其他城市的最短“路径”。

支持根据不同的优化目标（时间或费用）进行计算。

6.最优路径查询：

接受起始城市、终点城市和优化目标，调用Dijkstra算法进行计算，返回最优路径和相关信息（时间或费用）。