1. **车**

|  |  |  |
| --- | --- | --- |
| **Car** | | |
| **成员** | | |
| **int** | **id** |  |
| **int** | **from** |  |
| **int** | **to** |  |
| **int** | **max\_speed** |  |
| **int** | **cur\_speed** | **车在当前道路实际行驶速度** |
| **int** | **start\_time** |  |
| **int** | **position.roadId** | **当前所在道路** |
| **int** | **position.channel** | **当前所在道路数组的具体位置** |
| **int** | **position.pos** | **当前所在道路数组的具体位置** |
| **vector<int>** | **route** | **记录车辆当前规划路线(路口)** |
| **int** | **route\_idx** | **车辆当前行驶至规划路线的哪一段** |
| **bool** | **is\_started** | **汽车是否出发** |
| **bool** | **is\_arrived** | **汽车是否到达** |
| int | taffCount | 汽车在本次规划中被迫减速或堵车的次数统计（优化指标） |
| vector<int> | taffList | 汽车在本次规划中被迫减速或堵车的道路id（优化指标） |
| int | runTime | 汽车在本次规划中到达花费的总时间（优化指标） |
| int | direction | 记录汽车在下一路口的转向方向。2为直行，1为左转，0为右转。 |
| **方法** | | |
| **int** | **getRealSpeedOfRoad(Road& road);** | **返回该车在某条道路的行驶速度** |
| **int** | **getRunTimeOfRoad(Road& road);** | **返回该车在某条路上的行驶时间（整型，用于模拟路长）** |
| void | Update() | 将车子当前参数更新至最新 |
| void | SetVReal() | 根据当前路况计算车子的实际速度并更新vreal。 |

1. **路**

|  |  |  |
| --- | --- | --- |
| **Road** | | |
| **成员** | | |
| **int** | **id** |  |
| **int** | **length** |  |
| **int** | **from** |  |
| **int** | **to** |  |
| **int** | **max\_speed** |  |
| **int** | **channel** | **车道数** |
| **vector<vector<int>>** | **roadCondition** | **记录当前每条车道的实时路况，每个位置存放车辆id** |
| int | leftNum | 当前道路各车道总剩余车位 |
| int | taffFlow | 汽车在本次规划中流量统计（优化指标） |
| int | taffCount | 汽车在本次规划中被迫减速或堵车的次数统计（优化指标） |
| **方法** | | |
| void | Update() | 将当前道路参数更新至最新 |
| void | SetLeftNum() | 计算当前道路各车道总剩余车位（从最大车道开始累计）并更新leftNum |

1. **路口结点**

|  |  |  |
| --- | --- | --- |
| **Crossing** | | |
| 成员 | | |
| int | id |  |
| vector<int> | next\_cross |  |
| vector<int> | next\_road |  |
| int | pos.x | 路口结点在图中的相对位置 |
| int | pos.y | 路口结点在图中的相对位置 |
| bool | ispass | 用于进行BFS |
| int | time | 用于求最短路 |
| int | prev | 用于求最短路 |
| 方法 | | |
|  |  |  |

1. **图**

|  |  |  |
| --- | --- | --- |
| **Graph** | | |
| **成员** | | |
| **map<int, Cross>** | **cross\_map** | **路口id映射** |
| **map<int, Road>** | **road\_map** | **道路id映射** |
| **map<int, Car>** | **car\_map** | **汽车id映射** |
| **vector<int>** | **cross\_list** | **路口序列，存id** |
| **vector<int>** | **road\_list** | **道路序列，存id** |
| **vector<int>** | **car\_list** | **汽车序列，存id** |
| **map<pair<int, int>, int>** | **road\_direction** | **根据起始结尾结点对某条道路进行映射。** |
| **map<pair<int,int>, Cross>** | **rel\_coordinate\_map** | **用于绘制可视化地图** |
| **map<pair<int,int>, Cross>** | **abs\_coordinate\_map** | **用于绘制可视化地图** |
| **方法** | | |
| **void** | **initGraph()** | **根据输入数据生成路口的相对数据** |
| **void** | **initPath()** | **初始化所有车辆的最短路** |
| **void** | **getShortestRouteInSubmap(Car& car, vector<int>& submap)** | **在子图中求最短路** |
| **vector<int>** | **getSubMap(int start\_id, int end\_id);** | **根据起始与结尾结点求子图，返回所有子图路口id集合** |
| **void** | **modifyInitialPath()** | **对初始路径进行简单修正** |
| void | Update() | 将当前地图路况更新至最新 |
| void | Reset() | 重置所有汽车和道路的状态 |
| bool | IsFinish() | 判断是否所有汽车都已经抵达终点 |
| vector<int> | GetSubMap(int startid, int endid) | 输入起始和结束id，返回在两对角点范围矩形内的所有点集合 |
| void | Init() | 根据输入数据生成图并更新所有结点相对位置 |
| void | InitPath() | 为所有车生成最短路 |
| void | AmendPath() | 初步优化初始路径 |
| void | UpdateScheme(vector<int> amendcars) | 根据上一个方案的GetAmendCars()提供的车辆集合生成一个修正方案 |

1. **方案**

|  |  |  |
| --- | --- | --- |
| **Scheme** | | |
| **成员** | | |
| **vector<vector<int>>** | **answers** | **存储一个用于打印的回答，包括每辆车id和出发时间** |
| **vector<vector<int>>** | **cross\_routes** | **按路口记录的路径方案** |
| **vector<vector<int>>** | **road\_routes** | **按道路记录的路径方案** |
| **vector<vector<int>>** | **car\_runtimes** | **汽车行驶时间统计** |
| **vector<vector<int>>** | **car\_taffcounts** | **汽车堵车的次数统计** |
| **vector<vector<int>>** | **car\_taffroads** | **道路堵车次数统计** |
| **int** | **total\_whole\_time** | **方案花费总时间** |
| **int** | **toatal\_run\_time** | **汽车行驶总时间（最早出发至最晚到达）** |
| **int** | **total\_taff\_flow** | **道路总流量** |
| **int** | **total\_car\_taff\_count** | **堵车车辆总数** |
| **int** | **total\_road\_taff\_count** | **堵车道路总数** |
| 方法 | | |
| void | SetScheme(Map map) | 记录一次调度方案 |
| float | GetScore() | 按照自定义策略计算分数 |
| vector<int> | GetAmendCars() | 按自定义策略返回下一个迭代方案需要优化的车辆集合 |
| void | Print() | 按题目要求打印当前结果 |