

CNode

CNode类初始化

Parameters:

Name	Type	Description	Default
ID	int	节点的编号	None
zone_ID	int	区域ID	-1
PositionX	double	节点的X坐标	None
PositionY	double	节点的Y坐标	None
IncomingLink	vector<int>	进入节点的路段编号集合	None
OutgoingLink	vector<int>	离开节点的路段编号集合	None

CLink

CLink类初始化

Parameters:

Name	Type	Description	Default
ID	int	路段的编号，从0开始编号	None
InNodeIndex	int	路段起点的节点ID	None
OutNodeIndex	int	路段终点的节点ID	None
TravelTime	double	走行时间	None
Capacity	double	路段通行能力	9999
Alpha	double	BPR函数参数	0.15
Beta	double	BPR函数参数	4.0

CNetwork

CNetwork类初始化

Parameters:

Name	Type	Description	Default
m_nNode	int	节点数量	0
m_nLink	int	路段数量	0
m_Node	vector<CNode>	路网节点存储列表	None
m_Link	vector<CLink>	路网路段存储列表	None
MaxUEGap	double	允许最大误差	0.0001
LinkIndex	std::unordered_map<std::pair<int, int>, int, pair_hash>	起点和终点到边ID的映射表	None

CNetwork类方法 - AddEdgeFrom:

- 加入一条路段

Parameters:

Name	Type	Description	Default
input_tuple	tuple(int, int, dict{"weight":double})	三元元组：（起点，i终点, 字典(内有权重) ），必须参数	None

CNetwork类方法 - SingleSourceDijkstra:

- 单源最短路径迪杰斯特拉算法

Parameters:

Name	Type	Description	Default
start	int	起点，必须参数	None
end	int	终点，无输入时默认计算所有节点	None
cut_off	double	计算指定范围内的目标节点路径	inf

Returns:

没有输入end:

Name	Type	Description
{ShortestCost,ShortestPath}	tuple<unordered_map<int, double>, unordered_map<int, vector<int>>>>	{所有节点的最短路径总花费,所有节点的最短路径}

输入end:

Name	Type	Description
{ShortestCost,ShortestPath}	tuple<double,vector<int>>>	{目标节点的最短路径总花费,目标节点的最短路径}