

地图导航模拟算法。在一个无向图中任意给定两点实现最短路径计算。要求采用Dijkstra（深度或广度或Floyd或Bellman-Ford）算法，以及堆排序算法或其它排序算法，利用递归，vector支持邻接表方式，对于路径和选择路径进行界面显示（可仅仅显示而不交互）。主要目的：vector的理解，可视化编程的理解。

题目要求

后端要求

完成任意指定两点的最短路径计算，采用最短路算法，使用堆优化或其它排序算法，优化最短路算法，使用vector存图的邻接表。

前端要求

对于图的路径以及选择路径进行界面展示，可以仅仅显示但是不交互。

主要流程及解决思路

主要流程

- 1, 写一个小根堆。
- 2, 建立Vertex类，储存图的邻接表。
- 3, 建立一个堆存 $\text{pair} \langle \text{int}, \text{int} \rangle$ (cost, val) 存一个边集小根堆。
- 4, 利用堆写Dijkstra。
- 5, 建立EasyX界面，通过“建立地点”和“建立路线”或“文件读入”方式完成图的绘制。
- 6, 建立地点，在界面中以像素点进行标点，每次建点会向vector中进行一次push_back
- 7, 建立路线，输出起点和终点和边权，会完成邻接表的双向add，同时储存一个用于显示的边集。
- 8, 运行Dijkstra算法，计算路程，并把路线标红。

解决思路

要求使用简单的前端编程，对于C++，我已知的有EasyX和Qt。我在C语言课程设计中曾经使用过EasyX进行界面的设计，本次将继续使用EasyX完成相应的前端设计。要求使用求最短路的算法，因为题目要求，仅要求给定两点之间的最短路，使用Floyd等最短路算法，求各源最短路有歼星炮打蚊子的感觉，本次选择了求单元最短路的Dijkstra算法，堆优化后可以达到 $O(m \log n)$ 的级别，相比于Floyd的 $O(n^3)$ 要优不少。

程序的难点及遇到的问题

使用EasyX时，为了可以将控制按键与画面分开，进行了setorigin操作，但是鼠标的检测，不随着setorigin进行改变，导致出现无法鼠标操控的情况，进行了较长时间的debug。

程序的优缺点

使用了堆优化的Dijkstra算法，效率较优。使用鼠标控制，实现了简单的交互功能，边权的设置不仅仅通过两点之间的距离，提高了自由度，有利于进一步考察交通等各种因素。

我的收获

进一步掌握了有关EasyX的相关操作，对可视化编程有了进一步的理解。通过图形化的展示，对Dijkstra算法有了更深的理解。

代码实现

https://github.com/LeoMeng86/Cpp_experiment.git