# 行车路线

代码行数：130行

算法思想：将信息存储为邻接表。由于连续小道长度的不同，将点拆为不同的若干点。运用小项堆优化的 dijkstra对由1到不同点的最小疲劳值进行更新，同时用path和continuelength分别记录优化后节点对应上一个结点的数据（点标号和连续小路）。最后在每个点的所有拆点中找出最小的路径输出，对于n一次次向前（利用该节点存储的上一个结点的信息）推得出最短路径。

主要/核心函数分析：//堆优化版dijkstravoid dijkstra()初始化dist数组（储存不同的拆点x到1的最短疲惫值）。从1开始（此时1点的连续小路为0，最短疲惫值为0）对与它相连的点进行遍历，如果经过1的疲惫值比原先的要小，就更新dist对应的拆点，同时将该拆点的数据存入小项堆，依次取出堆中元素对dist进行更新，直到堆中没有元素。

测试数据：

6 7

1 1 2 3

1 2 3 2

0 1 3 30

0 3 4 20

0 4 5 30

1 3 5 6

1 5 6 1

运行结果：

到1最小疲劳值0

到2最小疲劳值9

到3最小疲劳值25

到4最小疲劳值45

到5最小疲劳值66

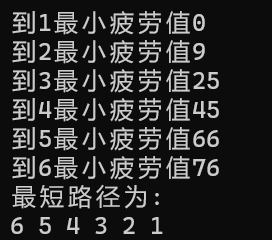
到6最小疲劳值76

最短路径为:

6 5 4 3 2 1

时间复杂度：O（边数）

结果截屏：



心得体会：熟悉了dijkstra的算法流程以及使用。同时学会了拆点，将x点根据连续小路的不同拆成很多点，更便于对疲劳值的计算与更新。同时在原有算法的基础上，学会运用数组记录前一个结点的信息，以便于输出路径。学会创建和使用大项堆和小项堆。同时还有邻接表用数组如何存储（以前是链表）。