一、调试成功程序及说明

1、

题目：

图的深度优先和广度优先遍历。

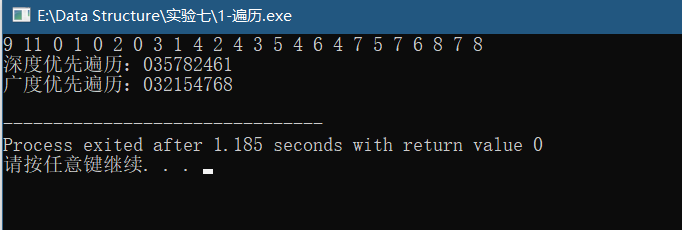
算法思想：

DFS：依次从每一个点出发，遍历至尽头；

BFS：依次从每一个点出发，遍历所有邻接点；

需要建立一个访问标记数组，防止重复遍历。

运行结果：



结果分析：O（n）

附源程序。

2、

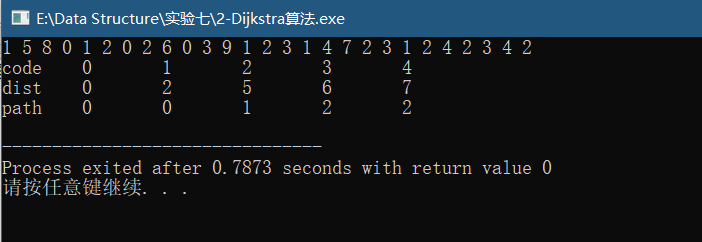
题目：

编程实现Dijkstra算法。

算法思想：

1. 初始化closet数组；
2. 选中距离最近的点；
3. 更新选中该点后的closet数组，直至全部选中。

运行结果：



结果分析：O（n^2）

附源程序。

3、

题目：

CSP题目

问题描述 ：Alice和Bob正在玩井字棋游戏。 井字棋游戏的规则很简单：两人轮流往3\*3的棋盘中放棋子，Alice放的是“X”，Bob放的是“O”，Alice执先。当同一种棋子占据一行、一列或一条对角线的三个格子时，游戏结束，该种棋子的持有者获胜。当棋盘被填满的时候，游戏结束，双方平手。   
　　Alice设计了一种对棋局评分的方法：   
　　- 对于Alice已经获胜的局面，评估得分为(棋盘上的空格子数+1)；   
　　- 对于Bob已经获胜的局面，评估得分为 -(棋盘上的空格子数+1)；   
　　- 对于平局的局面，评估得分为0；

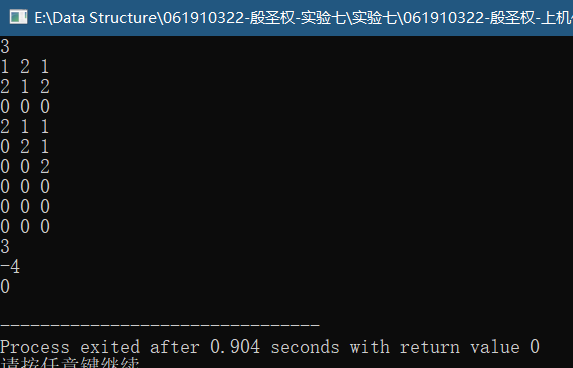
　　例如上图中的局面，Alice已经获胜，同时棋盘上有2个空格，所以局面得分为2+1=3。   
　　由于Alice并不喜欢计算，所以他请教擅长编程的你，如果两人都以最优策略行棋，那么当前局面的最终得分会是多少？   
输入格式   
　　输入的第一行包含一个正整数T，表示数据的组数。   
　　每组数据输入有3行，每行有3个整数，用空格分隔，分别表示棋盘每个格子的状态。0表示格子为空，1表示格子中为“X”，2表示格子中为“O”。保证不会出现其他状态。   
　　保证输入的局面合法。(即保证输入的局面可以通过行棋到达，且保证没有双方同时获胜的情况)   
　　保证输入的局面轮到Alice行棋。   
输出格式   
　　对于每组数据，输出一行一个整数，表示当前局面的得分。   
样例输入   
3   
1 2 1   
2 1 2   
0 0 0   
2 1 1   
0 2 1   
0 0 2   
0 0 0   
0 0 0   
0 0 0   
样例输出   
3   
-4   
0   
样例说明   
　　第一组数据：   
　　Alice将棋子放在左下角(或右下角)后，可以到达问题描述中的局面，得分为3。   
　　3为Alice行棋后能到达的局面中得分的最大值。   
　　第二组数据：

　　Bob已经获胜(如图)，此局面得分为-(3+1)=-4。   
　　第三组数据：   
　　井字棋中若双方都采用最优策略，游戏平局，最终得分为0。

算法思想：

计算空格数，再判断当前局面谁获胜，若A或B获胜，则直接计分；若没人获胜，则A下棋，下棋之后再判断A获胜还是A、B平局，然后计分。

运行结果：



结果分析：

O（n^2）

附源程序。

4、

题目：

CSP题目

问题描述：小刘承包了很多片麦田，为了灌溉这些麦田，小刘在第一个麦田挖了一口很深的水井，所有的麦田都从这口井来引水灌溉。 为了灌溉，小刘需要建立一些水渠，以连接水井和麦田，小刘也可以利用部分麦田作为“中转站”，利用水渠连接不同的麦田，这样只要一片麦田能被灌溉，则与其连接的麦田也能被灌溉。现在小刘知道哪些麦田之间可以建设水渠和建设每个水渠所需要的费用（注意不是所有麦田之间都可以建立水渠）。请问灌溉所有麦田最少需要多少费用来修建水渠。

输入格式：输入的第一行包含两个正整数n, m，分别表示麦田的片数和小刘可以建立的水渠的数量。麦田使用1, 2, 3, ……依次标号。 接下来m行，每行包含三个整数ai, bi, ci，表示第ai片麦田与第bi片麦田之间可以建立一条水渠，所需要的费用为ci。

输出格式：输出一个整数，表示灌溉所有麦田所需要的最小费用，及水渠连接说明。

**问题分析：**这个问题可以用最小生成树算法实现。

输入样例:

4 4

1 2 1

2 3 4

2 4 2

3 4 3

输出样例:

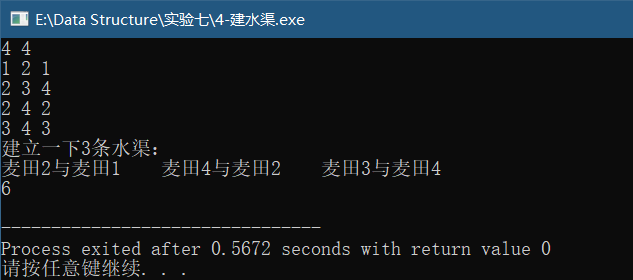
6

建立以下3条水渠：麦田1与麦田2、麦田2与麦田4、麦田4与麦田3。

算法思想：

应用prim算法，从麦田1出发求最小生成树。

运行结果：



结果分析：

O（n^2）

附源程序。

5、

题目：

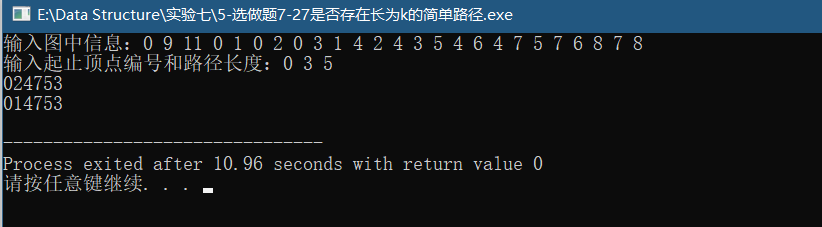
选做题7.27

采用邻接表存储结构，编写一个判别无向图中任意给定的两个顶点间是否存在一条长为k的简单路径的算法

算法思想：

基于DFS，若当前路径长度为k，则输出。

运行结果：



结果分析：

O（n）

附源程序。

6、

题目：

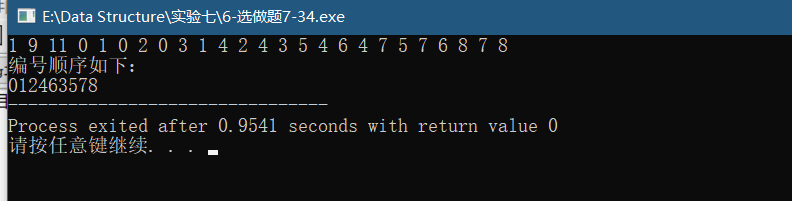
选做题7.34

试编写一个算法，给有向无环图G中每个顶点赋以一个整数序号，并满足：若从顶点i至顶点j有一条弧，则应使i<j。

算法思想：

进行一次拓扑排序，结果即为编号顺序。

运行结果：



结果分析：

O（n）

附源程序。

7、

题目：

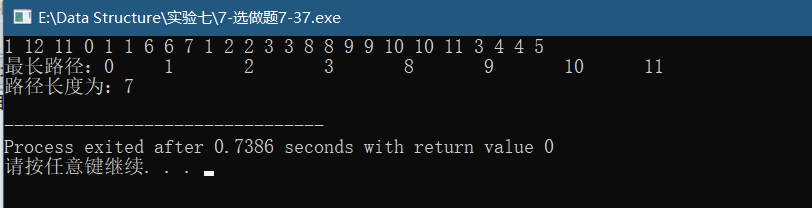
选做题7.37

设计一个求有向无环图最长路径的算法，并估计时间复杂度。

算法思想：

基于DFS，遍历每一条路径，若比已知的最长路径大，则替换。

运行结果：



结果分析：

O（n^2）

附源程序。

二、代码行数及小结

代码行数：

1、125

2、88

3、88

4、87

5、100

6、114

7、121

小结：

1. 对DFS和BFS有了深刻的理解，并能够加以应用；
2. 掌握了最小生成树、最短路径、关键路径等算法，并能够加以应用。