**《数据结构》上机实验题目**

**（共8次，每次上机4小时）**

**第一阶段（线性部分）**

**《数据结构》第1次上机题 （线性表练习）**

1．编程实现书P12 ADT List 基本操作14个：

* 1. 用顺序存储结构实现； （2）用链式存储结构实现；

1. 设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写函数，实现线性表的就地逆置（书P31 4）。
2. 设线性表L，元素值为整型的且存在相同值，分别采用顺序结构和链式结构存储，编写函数，利用原空间，删除重复的元素值。

4. CSP 题目

问题描述：一次放学的时候，小明已经规划好了自己回家的路线，并且能够预测经过各个路段的时间。同时，小明通过学校里安装的“智慧光明”终端，看到了**出发时刻路上**经过的所有红绿灯的指示状态。请帮忙计算小明此次回家所需要的时间。

输入格式：

　　输入的第一行包含空格分隔的三个正整数 r、y、g，表示红绿灯的设置。这三个数均不超过 106。  
　　输入的第二行包含一个正整数 n，表示小明总共经过的道路段数和路过的红绿灯数目。  
　　接下来的 n 行，每行包含空格分隔的两个整数 k、t。k=0 表示经过了一段道路，将会耗时 t 秒，此处 t 不超过 106；k=1、2、3 时，分别表示**出发时刻**，此处的红绿灯状态是红灯、黄灯、绿灯，且倒计时显示牌上显示的数字是 t，此处 t 分别不会超过 r、y、g。

输出格式：

　　输出一个数字，表示此次小明放学回家所用的时间。

样例输入：

30 3 30  
8  
0 10  
1 5  
0 11  
2 2  
0 6  
0 3  
3 10  
0 3

样例输出：

46

样例说明：

　　小明先经过第一段路，用时10秒。第一盏红绿灯出发时是红灯，还剩5秒；小明到达路口时，这个红绿灯已经变为绿灯，不用等待直接通过。接下来经过第二段路，用时11秒。第二盏红绿灯出发时是黄灯，还剩两秒；小明到达路口时，这个红绿灯已经变为红灯，还剩11秒。接下来经过第三、第四段路，用时9秒。第三盏红绿灯出发时是绿灯，还剩10秒；小明到达路口时，这个红绿灯已经变为红灯，还剩两秒。接下来经过最后一段路，用时3秒。共计10+11+11+9+2+3=46秒。

评测用例规模与约定：

　　有些测试点具有特殊的性质：  
　　\* 前2个测试点中不存在任何信号灯。  
　　测试点的输入数据规模：  
　　\* 前6个测试点保证n≤103。  
　　\* 所有测试点保证n≤105。

5. CSP 题目

问题描述：近来，跳一跳这款小游戏风靡全国，受到不少玩家的喜爱。  
　　简化后的跳一跳规则如下：玩家每次从当前方块跳到下一个方块，如果没有跳到下一个方块上则游戏结束。  
　　如果跳到了方块上，但没有跳到方块的中心则获得1分；跳到方块中心时，若上一次的得分为1分或这是本局游戏的第一次跳跃则此次得分为2分，否则此次得分比上一次得分多两分（即连续跳到方块中心时，总得分将+2，+4，+6，+8...）。  
　　现在给出一个人跳一跳的全过程，请你求出他本局游戏的得分（按照题目描述的规则）。

输入格式：

　　输入包含多个数字，用空格分隔，每个数字都是1，2，0之一，1表示此次跳跃跳到了方块上但是没有跳到中心，2表示此次跳跃跳到了方块上并且跳到了方块中心，0表示此次跳跃没有跳到方块上（此时游戏结束）。

输出格式：

　　输出一个整数，为本局游戏的得分（在本题的规则下）。

样例输入：

1 1 2 2 2 1 1 2 2 0

样例输出：

22

数据规模和约定：

　　对于所有评测用 例，输入的数字不超过30个，保证0正好出现一次且为最后一个数字。

附加题：习题集 1.19 1.20 2.19

**《数据结构》第2次上机题 （线性表练习）**

1. 设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写函数，用选择/冒泡排序算法实现线性表的表排序。

2．设线性表A、B，元素值为整型，且递减有序，编写函数，实现下列功能：对采用顺序结构和链式结构2种存储结构，要求在A的空间上构成一个新线性表C，其元素为A和B元素的并集，且表C中的元素值递减有序（互不相同）。

3．输入正整数n、m（m<n），设有n个人坐成一圈，从第1个人开始循环报数，报到m的人出列，然后再从下一个人开始报数，报到m的人又出列，如此重复，直到所有的人都出列为止。要求用链式结构和顺序结构实现，按出列的先后顺序输出每个人的信息。

4．CSP题目

题目描述：小明在他的果园里种了一些苹果树，这些苹果树排列成一个圆。为了保证苹果的品质，在种植过程中要进行疏果操作。为了更及时地完成疏果操作，小明会不时地检查每棵树的状态，根据需要进行疏果。检查时，如果发现可能有苹果从树上掉落，小明会重新统计树上的苹果个数（然后根据之前的记录就可以判断是否有苹果掉落了)。在全部操作结束后，请帮助小明统计相关的信息。

输入格式：

从标准输入读入数据。

第1行包含一个正整数N，表示苹果树的棵数。

第1+i行（1≤i≤N)，每行的格式为mi,ai1,ai2,... ,aimi,。其中，第一个正整数mi表示本行后面的整数个数。后续的mi个整数表示小明对第i棵苹果树的操作记录。若aij (1≤ j ≤mi）为正整数，则表示小明进行了重新统计该棵树上的苹果个数的操作，统计的苹果个数为aij;若为零或负整数，则表示一次疏果操作，去掉的苹果个数是|aij|。

输入保证一定是正确的，满足:

1. ai1>0，即对于每棵树的记录，第一个操作一定是统计苹果个数（初始状态，此时不用判断是否有苹果掉落）；
2. 每次疏果操作保证操作后树上的苹果个数仍为正。

输出格式：

输出到标准输出。

输出只有一行，包含三个整数T、D、E。其中，

1. T为全部疏果操作结束后所有苹果树上剩下的苹果总数（假设每棵苹果树在最后一次统计苹果个数操作后苹果不会因为疏果以外的原因减少);
2. D为发生苹果掉落的苹果树的棵数;
3. E为相邻连续三棵树发生苹果掉落情况的组数。

对于第三个统计量的解释：N棵苹果树A1,A2,...,AN排列成一个圆，那么A1与A2相邻，A2与A3相邻，......，AN-1与AN相邻，AN与A1相邻。如果Ai-1, Ai，Ai+1这三棵树都发生了苹果掉落的情况，则记为一组。形式化的，有

E=|{Ai/Drop(Pred(Ai)）∧Drop(Ai）∧Drop(Succ(Ai)）}|

其中，Drop(Ai）表示苹果树Ai是否发生苹果掉落的情况，Pred(Ai)表示Ai的前一棵树Ai-1（如果i>1）或者AN（如果i=1)，Succ(Ai)表示Ai的后一棵树Ai+1（如果i<N）或者A1（如果i=N）

样例1输入：

4

4 74 -7 -12 -5

5 73 -8 -6 59 -4

5 76 -5 -10 60 -2

5 80 -6 -15 59 0

样例1输出：

222 1 0

样例1解释：

全部操作结束后，第1棵树上剩下的苹果个数为74-7-12-5=50，第2棵为59-4=55，第3棵为60-2=58，第4棵为59-0=59。因此T=50+55+58+59=222。

其中，第3棵树在第2次统计之前剩下的苹果个数为76-5-10=61>60，因此发生了苹果掉落的情况。可以检验其他的树没有这种情况，因此D=1。

没有连续三棵树都发生苹果掉落的情况，因此E=0。

样例2输入：

5

4 10 0 9 0

4 10 -2 7 0

2 10 0

4 10 -3 5 0

4 10 -1 8 0

样例2输出：

39 4 2

样例2解释：

第1、2、4、5棵树发生了苹果掉落的情况，因此D=4。其中，连续三棵树都发生苹果掉落情况的有(5,1,2)和(4,5,1)，因此E=2。

5．CSP题目

问题描述：小H和小W来到了一条街上，两人分开买菜，他们买菜的过程可以描述为，去店里买一些菜然后去旁边的一个广场把菜装上车，两人都要买n种菜，所以也都要装n次车。具体的，对于小H来说有n个不相交的时间段[a1,b1],[a2,b2]…[an,bn]在装车，对于小W来说有n个不相交的时间段[c1,d1],[c2,d2]…[cn,dn]在装车。其中，一个时间段[s, t]表示的是从时刻s到时刻t这段时间，时长为t-s。  
　　由于他们是好朋友，他们都在广场上装车的时候会聊天，他们想知道他们可以聊多长时间。  
输入格式  
　　输入的第一行包含一个正整数n，表示时间段的数量。  
　　接下来n行每行两个数ai，bi，描述小H的各个装车的时间段。  
　　接下来n行每行两个数ci，di，描述小W的各个装车的时间段。  
输出格式  
　　输出一行，一个正整数，表示两人可以聊多长时间。  
样例输入  
4  
1 3  
5 6  
9 13  
14 15  
2 4  
5 7  
10 11  
13 14  
样例输出  
3  
数据规模和约定  
　　对于所有的评测用例，1 ≤ n ≤ 2000, ai < bi < ai+1，ci < di < ci+1,对于所有的i(1 ≤ i ≤ n)有，1 ≤ ai, bi, ci, di ≤ 1000000。  
　　给两个人设定两个数组t[i]，t1[i] 当装车时置1，当t[i]=t1[i]时 总数sum++。

附加题：习题集 2.24 2.29 2.30

**《数据结构》 第3次上机题（线性表复习，栈与队列练习）**

1．编程实现书P32 ADT Stack 基本操作9个，用顺序存储结构实现；

2．编程实现书P48 ADT Queue 基本操作9个，用链式存储结构实现；

3. 利用栈操作实现八皇后问题求解。

**4.** CSP题目

问题描述：小明今天生日，他有n块蛋糕要分给朋友们吃，这n块蛋糕（编号为1到n）的重量分别为a1, a2, …, an。小明想分给每个朋友至少重量为k的蛋糕。小明的朋友们已经排好队准备领蛋糕，对于每个朋友，小明总是先将自己手中编号最小的蛋糕分给他，当这个朋友所分得蛋糕的重量不到k时，再继续将剩下的蛋糕中编号最小的给他，直到小明的蛋糕分完或者这个朋友分到的蛋糕的总重量大于等于k。  
　　请问当小明的蛋糕分完时，总共有多少个朋友分到了蛋糕。

输入格式：

　　输入的第一行包含了两个整数n, k，意义如上所述。  
　　第二行包含n个正整数，依次表示a1, a2,…, an。

输出格式：

　　输出一个整数，表示有多少个朋友分到了蛋糕。

样例输入：

6 9  
2 6 5 6 3 5

样例输出：

3

样例说明：

　　第一个朋友分到了前3块蛋糕，第二个朋友分到了第4、5块蛋糕，第三个朋友分到了最后一块蛋糕。

评测用例规模与约定：

　　对于所有评测用例，1≤n≤1000，1≤k≤10000，1≤ai≤1000。

5. CSP题目

问题描述：给定n个数，请找出其中相差（差的绝对值）最小的两个数，输出它们的差值的绝对值。

输入格式：

　　输入第一行包含一个整数n。  
　　第二行包含n个正整数，相邻整数之间使用一个空格分隔。

输出格式：

　　输出一个整数，表示答案。

样例输入：

5  
1 5 4 8 20

样例输出：

1

样例说明：

　　相差最小的两个数是5和4，它们之间的差值是1。

样例输入：

5  
9 3 6 1 3

样例输出：

0

样例说明：

　　有两个相同的数3，它们之间的差值是0.

数据规模和约定：

　　对于所有评测用例，2≤n≤1000，每个给定的整数都是不超过10000的正整数。

附加题：习题集 2.32 2.37 2.38

**《数据结构》 第4次上机题（线性结构练习）**

1．输入稀疏矩阵，建立稀疏矩阵三元组顺序结构，实现矩阵的列序遍历转置和快速转置算法。

2. 求矩阵的马鞍点。（书P69 7）

3. CSP题目

题目背景：某地疫情爆发后，出于“应检尽检”的原则，我们想要通知所有近期经过该高危区域的居民参与核酸检测。

问题描述：想要找出经过高危区域的居民，分析位置记录是一种简单有效的方法。

具体来说，一位居民的位置记录包含t个平面坐标(x1,y1),(x2,y2),…,(xt,yt), 其中(xi,yi) 表示该居民i时刻所在位置。

高危区域则可以抽象为一个矩形区域（含边界），左下角和右上角的坐标分别为(xl,yd)和(xr,yu)，满足xl<xr且yd<yu。

考虑某位居民的位置记录，如果其中某个坐标位于矩形内（含边界），则说明该居民经过高危区域；进一步地，如果其中连续k个或更多坐标均位于矩形内（含边界），则认为该居民曾在高危区域逗留。需要注意的是，判定经过和逗留时我们只关心位置记录中的t个坐标，而无需考虑该居民在i到i+1时刻之间位于何处。

给定高危区域的范围和n位居民过去t个时刻的位置记录，试统计其中经过高危区域的人数和曾在高危区域逗留的人数。

输入格式：

输入共n+1行。

第一行包含用空格分隔的七个整数n、k、t、xl、yd、xr和yu，含义如上文所述。

接下来n行，每行包含用空格分隔的2t个整数，按顺序表示一位居民过去t个时刻的位置记录(x1,y1),(x2,y2),…,(xt,yt)。

输出格式：

输出共两行，每行一个整数，分别表示经过高危区域的人数和曾在高危区域逗留的人数。

样例输入1：

5 2 6 20 40 100 80

100 80 100 80 100 80 100 80 100 80 100 80

60 50 60 46 60 42 60 38 60 34 60 30

10 60 14 62 18 66 22 74 26 86 30 100

90 31 94 35 98 39 102 43 106 47 110 51

0 20 4 20 8 20 12 20 16 20 20 20

样例输出1：

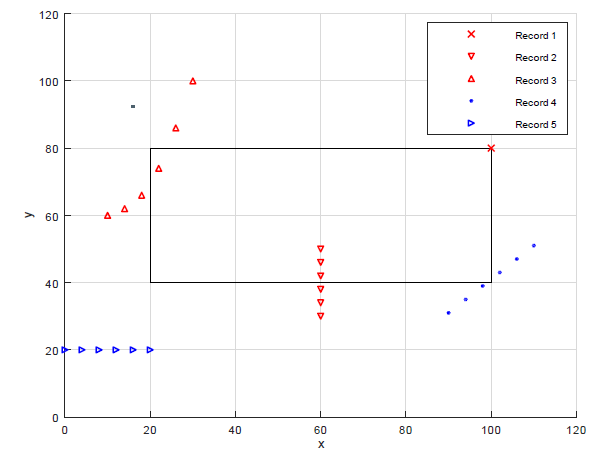
3

2

样例1说明：

如下图红色标记所示，前三条位置记录经过了高危区域；

但第三条位置记录（图中左上曲线）只有一个时刻位于高危区域内，不满足逗留条件。



样例输入2：

1 3 8 0 0 10 10

-1 -1 0 0 0 0 -1 -1 0 0 -1 -1 0 0 0 0

样例输出2：

1

0

样例2说明：

该位置记录经过了高危区域，但最多只有连续两个时刻位于其中，不满足逗留条件。

评测用例规模与约定：

全部的测试点满足1≤n≤20，1≤k≤t≤103，所有坐标均为整数且绝对值不超过106。

4. CSP题目

问题描述：请实现一个铁路购票系统的简单座位分配算法，来处理一节车厢的座位分配。  
　　假设一节车厢有20排、每一排5个座位。为方便起见，我们用1到100来给所有的座位编号，第一排是1到5号，第二排是6到10号，依次类推，第20排是96到100号。  
　　购票时，一个人可能购一张或多张票，最多不超过5张。如果这几张票可以安排在同一排编号相邻的座位，则应该安排在编号最小的相邻座位。否则应该安排在编号最小的几个空座位中（不考虑是否相邻）。  
　　假设初始时车票全部未被购买，现在给了一些购票指令，请你处理这些指令。

输入格式：对于所有评测用例，1 ≤ n ≤ 100，所有购票数量之和不超过100。

　　输入的第一行包含一个整数n，表示购票指令的数量。  
　　第二行包含n个整数，每个整数p在1到5之间，表示要购入的票数，相邻的两个数之间使用一个空格分隔。

输出格式

输出n行，每行对应一条指令的处理结果。  
　　对于购票指令p，输出p张车票的编号，按从小到大排序。

**问题分析：**这个问题可以用顺序结构或链式结构实现。

样例输入

4  
2 5 4 2

样例输出

1 2  
6 7 8 9 10  
11 12 13 14  
3 4

附加题：习题集 3.20 3.28 3.32

**第一阶段总结**

**第二阶段（树与图部分）**

**《数据结构》 第5次上机题目 (二叉树练习 )**

1. 编程实现书P75 ADT BinaryTree 基本操作20个，用二叉链表结构实现；

2.实现二叉树的先序、中序、后序遍历，用递归和非递归方法；实现层次遍历。

3. 设二叉树采用二叉链表存储，编写函数，对二叉树中每个元素值为x的结点，删除以它为根的子树，并释放相应空间。（习题集6.45）

4. 编写函数，判断给定的二叉树是否是完全二叉树。（习题集6.49）

5． CSP题目

题目描述：甲乙丙丁决定玩一个报数的游戏来打发时间。游戏规则为四个人从1开始轮流进行报数，但如果需要报出的数是7的倍数或含有数字7则直接跳过。

此外大家约定，在总共报出了n个数后（不计入被跳过的数）游戏结束。现在需要你来帮忙统计，游戏过程中每个人各自跳过了几次。

输入格式：

从标准输入读入数据。

输入仅一行，包含一个正整数n，表示报出了多少个数后游戏结束。

输出格式：

输出到标准输出。

输出共四行，每行一个整数，依次表示甲乙丙丁四人在游戏过程中跳过的次数。

样例1输入：

20

样例1输出：

2

1

1

0

样例1解释：

报数过程为:

甲：1，乙：2，丙：3，丁：4

甲：5，乙：6，丙：跳过，丁：8

甲：9，乙：10，丙：11，丁：12

甲：13，乙：跳过，丙：15，丁：16

甲：跳过，乙：18，丙：19，丁：20

甲：跳过，乙：22，丙：23，丁：24

在丁报出24后，四个人总计报出了20个数，游戏结束。

样例2输入：

66

样例2输出：

7

5

11

5

评测用例规模与约定：

保证n≤666。

附加题：习题集 6.42 6.45 6.47

**《数据结构》 第6次上机题目 ( 二叉树、树、图练习 )**

1．编程实现书P96 ADT Graph 基本操作11个，用邻接矩阵存储结构实现；

2. 输入N个权值（1-100正整数），建立哈夫曼树。

3. 编写函数，对二叉链表结构的二叉树，求宽度。（书P94 4）

4. 编写函数，对一棵以孩子-兄弟链表表示的树，输出第i层的所有元素。

5. CSP题目

问题描述：俄罗斯方块是俄罗斯人阿列克谢·帕基特诺夫发明的一款休闲游戏。

　　游戏在一个15行10列的方格图上进行，方格图上的每一个格子可能已经放置了方块，或者没有放置方块。每一轮，都会有一个新的由4个小方块组成的板块从方格图的上方落下，玩家可以操作板块左右移动放到合适的位置，当板块中某一个方块的下边缘与方格图上的方块上边缘重合或者达到下边界时，板块不再移动，如果此时方格图的某一行全放满了方块，则该行被消除并得分。

　　在这个问题中，你需要写一个程序来模拟板块下落，你不需要处理玩家的操作，也不需要处理消行和得分。

　　具体的，给定一个初始的方格图，以及一个板块的形状和它下落的初始位置，你要给出最终的方格图。

输入格式：

　　输入的前15行包含初始的方格图，每行包含10个数字，相邻的数字用空格分隔。如果一个数字是0，表示对应的方格中没有方块，如果数字是1，则表示初始的时候有方块。输入保证前4行中的数字都是0。

　　输入的第16至第19行包含新加入的板块的形状，每行包含4个数字，组成了板块图案，同样0表示没方块，1表示有方块。输入保证板块的图案中正好包含4个方块，且4个方块是连在一起的（准确的说，4个方块是四连通的，即给定的板块是俄罗斯方块的标准板块）。

　　第20行包含一个1到7之间的整数，表示板块图案最左边开始的时候是在方格图的哪一列中。注意，这里的板块图案指的是16至19行所输入的板块图案，如果板块图案的最左边一列全是0，则它的左边和实际所表示的板块的左边是不一致的（见样例）

输出格式：

输出15行，每行10个数字，相邻的数字之间用一个空格分隔，表示板块下落后的方格图。注意，你不需要处理最终的消行。

样例输入：

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 1 0 0 0

1 1 1 0 0 0 1 1 1 1

0 0 0 0 1 0 0 0 0 0

0 0 0 0

0 1 1 1

0 0 0 1

0 0 0 0

3

样例输出：

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 1 0 0 0

1 1 1 1 1 1 1 1 1 1

0 0 0 0 1 1 0 0 0 0

附加题：习题集 6.49 6.60 6.62

**《数据结构》 第7次上机题目 ( 图 练习 )**

1. 图的深度优先和广度优先遍历；

2．编程实现Dijkstra算法；

3．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。

4．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。

附加题：习题集 7.27 7.34 7.37

**第二阶段总结**

**第三阶段（查找与排序部分）**

**《数据结构》 第8次上机题目 ( 查找 排序 练习 )**

1. 实现二叉排序树的插入和删除。

2．实现交换、选择、归并等简单排序算法；

3．实现快速排序算法；

4．实现堆排序算法；

5．CSP题目

题目背景：

开学了，可是校园里堆积了不少垃圾杂物。

热心的同学们纷纷自发前来清理，为学校注入正能量～

题目描述：

通过无人机航拍我们已经知晓了n处尚待清理的垃圾位置，其中第i (1≤i≤n)处的坐标为(xi,yi)，保证所有的坐标均为整数。

我们希望在垃圾集中的地方建立些回收站。具体来说，对于一个位置(x,y)是否适合建立回收站，我们主要考虑以下几点：

1. (x,y)必须是整数坐标，且该处存在垃圾；
2. 上下左右四个邻居位置，即(x,y+1)、(x,y-1)、(x+1,y)和(x-1,y)处，必须全部存在垃圾；
3. 进一步地，我们会对满足上述两个条件的选址进行评分分数为不大于4的自然数，表示在(x±1,y±1)四个对角位置中有几处存在垃圾。

现在，请你统计一下每种得分的选址个数。

输入格式：

从标准输入读入数据。

输入总共有n+1行。

第1行包含一个正整数n，表示已查明的垃圾点个数。

第1+i行（1≤i≤n）包含由一个空格分隔的两个整数xi和yi，表示第i处垃圾的坐标。

保证输入的n个坐标互不相同。

输出格式：

输出到标准输出。

输出共五行，每行一个整数，依次表示得分为0、1、2、3和4的回收站选址个数。

样例1输入：

7

1 2

2 1

0 0

1 1

1 0

2 0

0 1

样例1输出：

0

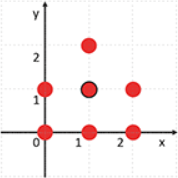
0

1

0

0

样例1解释：



如图所示，仅有(1,1)可选为回收站地址，评分为2。

样例2输入：

2

0 0

-100000 10

样例2输出：

0

0

0

0

0

样例2解释：

不存在可选地址。

样例3输入：

11

9 10

10 10

11 10

12 10

13 10

11 9

11 8

12 9

10 9

10 11

12 11

样例3输出：

0

2

1

0

0

样例3解释：

1分选址：(10,10)和(12,10);

2分选址：(11,9)。

提示：

本题中所涉及的坐标皆为整数，且保证输入的坐标两两不同。

附加题：习题集 9.32 10.32 10.34

**第三阶段总结**