《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 上机题 （线性表练习）

1．编程实现书 P19 ADT List 基本操作 12 个：

（1） 用顺序存储结构实现； （2）用链式存储结构实现；

2．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，编写程序，实现线性表的就地逆置（习题集 P18 2.21， 2.22）。

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 学生排队

问题描述

体育老师小明要将自己班上的学生按顺序排队。他首先让学生按学号从小到大的顺序排成一排，学号小的排在前面，然后进行多次调整。一次调整小明可能让一位同学出队，向前或者向后移动一段距离后再插入队列。

例如，下面给出了一组移动的例子，例子中学生的人数为 8 人。

0）初始队列中学生的学号依次为 1, 2, 3, 4, 5, 6, 7, 8；

1）第一次调整，命令为“3 号同学向后移动 2”，表示 3 号同学出队，向后移动 2 名同学的距离，再插入到队列中，新队列中学生的学号依次为 1, 2, 4, 5, 3, 6, 7, 8；

2）第二次调整，命令为“8 号同学向前移动 3”，表示 8 号同学出队，向前移动 3 名同学的距离，再插入到队列中，新队列中学生的学号依次为 1, 2, 4, 5, 8, 3, 6, 7；

3）第三次调整，命令为“3 号同学向前移动 2”，表示 3 号同学出队，向前移动 2 名同学的距离，再插入到队列中，新队列中学生的学号依次为 1, 2, 4, 3, 5, 8, 6, 7。

小明记录了所有调整的过程，请问，最终从前向后所有学生的学号依次是多少？

请特别注意，上述移动过程中所涉及的号码指的是学号，而不是在队伍中的位置。在向后移动时，移动的距离不超过对应同学后面的人数，如果向后移动的距离正好等于对应同学后面的人数则该同学会移动到队列的最后面。在向前移动时，移动的距离不超过对应同学前面的人数，如果向前移动的距离正好等于对应同学前面的人数则该同学会移动到队列的最前面。

输入格式

输入的第一行包含一个整数 n，表示学生的数量，学生的学号由 1 到 n 编号。

第二行包含一个整数 m，表示调整的次数。

接下来 m 行，每行两个整数 p，q，如果 q 为正，表示学号为 p 的同学向后移动 q，如果 q 为负，表示学号为 p 的同学向前移动-q。

输出格式

输出一行，包含 n 个整数，相邻两个整数之间由一个空格分隔，表示最终从前向后所有学生的学号。

样例输入

8

3

3 2

8 -3

3 -2

样例输出

1 2 4 3 5 8 6 7

问题分析：这个问题可以通过顺序结构或（双向或单向）链表实现，但对于移动元素较多的情况，应采用哪种存储结构更优呢？

选做题：习题集 1.19 1.20 2.19

《数据结构》第 2 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，编写程序，用插入排序算法实现线性表的表排序。

2．设元素值为整型的线性表 A,、B，分别采用顺序结构和链式结构存储，编写程序，实现下列功能：假设以递增有序排列的线性表 A 和 B 分别表示两个集合，现要求在 A 的空间上构成一个新线性表 C，其元素为 A 和 B 元素的交集，且表 C 中的元素也是依值递增有序排列。

3．CSP 题目

问题描述：先输入一个十进制整数 n，再输入 n 个正整数，求它们相邻数之差可知是否为上升或下降，由上升转下降或由下降转上升为折点，求折点数。

问题分析：如果一个点的值比左右两个都大或都小，则为折点。

样例输入：

5

1 3 5 2 1

样例输出

1

样例输入：

6

3 5 1 7 8 4

样例输出

3

4．CSP 题目

问题描述：首先输入正整数 n，接着输入 n 个正整数，如果其中存在一个数，比该数大的个数等于比该数小的个数，则输出该数，如果不存在则输出-1。

问题分析：这个问题可以用排序来实现。

样例输入：

6

9 7 2 5 7 8

样例输出

7

样例输入：

5

9 5 2 5 7

样例输出

-1

选做题：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 P45 ADT Stack 基本操作 9 个，用顺序存储结构实现；

2．编程实现书 P59 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

在某图形操作系统中,有 N 个窗口,每个窗口都是一个两边与坐标轴分别平行的矩形区域。窗口的边界上的点也属于该窗口。窗口之间有层次的区别,在多于一个窗口重叠的区域里,只会显示位于顶层的窗口里的内容。

当你点击屏幕上一个点的时候,你就选择了处于被点击位置的最顶层窗口,并且这个窗口就会被移到所有窗口的最顶层,而剩余的窗口的层次顺序不变。如果你点击的位置不属于任何窗口,则系统会忽略你这次点击。

现在我们希望你写一个程序模拟点击窗口的过程。

输入格式

输入的第一行有两个正整数,即 N 和 M。(1 ≤ N ≤ 10,1 ≤ M ≤ 10)

接下来 N 行按照从最下层到最顶层的顺序给出 N 个窗口的位置。 每行包含四个非负整数 x_1, y_1, x_2, y_2,表示该窗口的一对顶点坐标分别为 (x_1, y_1) 和 (x_2, y_2)。保证 $x_1 < x_2$, $y_1 < y_2$。

接下来 M 行每行包含两个非负整数 x, y,表示一次鼠标点击的坐标。

题目中涉及到的所有点和矩形的顶点的 x, y 坐标分别不超过 2559 和 1439。

问题分析：这个问题可以用链式线性表来实现。

输出格式：输出包括 M 行,每一行表示一次鼠标点击的结果。如果该次鼠标点击选择了一个窗口,则输出这个窗口的编号(窗口按照输入中的顺序从 1 编号到 N);如果没有,则输出"IGNORED"(不含双引号)。

样例输入

3 4

0 0 4 4

1 1 5 5

2 2 6 6

1 1

0 0

4 4

0 5

样例输出

2

1

1

IGNORED

样例说明

第一次点击的位置同时属于第 1 和第 2 个窗口,但是由于第 2 个窗口在上面,它被选择并且被置于顶层。

第二次点击的位置只属于第 1 个窗口,因此该次点击选择了此窗口并将其置于顶层。现在的三个窗口的层次关系与初始状态恰好相反了。

第三次点击的位置同时属于三个窗口的范围,但是由于现在第 1 个窗口处于顶层,它被选择。

最后点击的 (0, 5) 不属于任何窗口。

选做题：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，实现转置（1、2）；

2. CSP 题目

问题描述:首先输入正整数 n(n<10000),接着输入 n 个正整数(最大值为 10000),对于这 n 个数，统计输出其中的相邻数对（差值为 1 的数对），相同数据只被统计一次。

问题分析：这个看似是一个 n 个数与 n 个数进行比较（$O(n^2)$）的问题,能否用高效的方法解决？

样例输入

6

1 3 8 2 5 2

样例输出

2

样例输入

5

4 3 6 3 5 2

样例输出

4

3. 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

选做题：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 P121 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. 编程实现，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。

4. CSP 题目 消除类游戏

问题描述

消除类游戏是深受大众欢迎的一种游戏，游戏在一个包含有 n 行 m 列的游戏棋盘上进行，棋盘的每一行每一列的方格上放着一个有颜色的棋子，当一行或一列上有连续三个或更多的相同颜色的棋子时，这些棋子都被消除。当有多处可以被消除时，这些地方的棋子将同时被消除。

现在给你一个 n 行 m 列的棋盘，棋盘中的每一个方格上有一个棋子，请给出经过一次消除后的棋盘。

请注意：一个棋子可能在某一行和某一列同时被消除。

输入格式

输入第一行包含两个整数 n, m，用空格分隔，分别表示棋盘的行数和列数。

接下来 n 行，每行 m 个整数，用空格分隔，分别表示每一个方格中的棋子的颜色。颜色使用 1 至 9 编号。

输出格式

输出 n 行，每行 m 个整数，相邻的整数之间使用一个空格分隔，表示经过一次消除后的棋盘。如果一个方格中的棋子被消除，则对应的方格输出 0，否则输出棋子的颜色编号。

样例输入

4 5

2 2 3 1 2

3 4 5 1 4

2 3 2 1 3

2 2 2 4 4

样例输出

2 2 3 0 2

3 4 5 0 4

2 3 2 0 3

0 0 0 4 4

样例说明

棋盘中第 4 列的 1 和第 4 行的 2 可以被消除，其他的方格中的棋子均保留。

样例输入

4 5

2 2 3 1 2

3 1 1 1 1

2 3 2 1 3

2 2 3 3 3

样例输出

2 2 3 0 2

3 0 0 0 0

2 3 2 0 3

2 2 0 0 0

样例说明

棋盘中所有的 1 以及最后一行的 3 可以被同时消除，其他的方格中的棋子均保留。

问题分析：这个问题与树无关，可以使用二维数组来存储，通过一遍遍历对符合条件的格子进行标记，然后第二遍遍历时消除符合条件的格子。

选做题：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 P156 ADT Graph 基本操作 13 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. 编程实现，对一棵以孩子-兄弟链表表示的树，求树的高度。
4. CSP 题目 碰撞的小球

问题描述

数轴上有一条长度为 L（L 为偶数)的线段，左端点在原点，右端点在坐标 L 处。有 n 个不计体积的小球在线段上，开始时所有的小球都处在偶数坐标上，速度方向向右，速度大小为 1 单位长度每秒。

当小球到达线段的端点（左端点或右端点）的时候，会立即向相反的方向移动，速度大小仍然为原来大小。

当两个小球撞到一起的时候，两个小球会分别向与自己原来移动的方向相反的方向，以原来的速度大小继续移动。

现在，告诉你线段的长度 L，小球数量 n，以及 n 个小球的初始位置，请你计算 t 秒之后，各个小球的位置。

提示

因为所有小球的初始位置都为偶数，而且线段的长度为偶数，可以证明，不会有三个小球同时相撞，小球到达线段端点以及小球之间的碰撞时刻均为整数。

同时也可以证明两个小球发生碰撞的位置一定是整数（但不一定是偶数)。

输入格式

输入的第一行包含三个整数 n, L, t，用空格分隔，分别表示小球的个数、线段长度和你需要计算 t 秒之后小球的位置。

第二行包含 n 个整数 a1，a2，…，an，用空格分隔，表示初始时刻 n 个小球的位置。

输出格式

输出一行包含 n 个整数，用空格分隔，第 i 个整数代表初始时刻位于 ai 的小球，在 t 秒之后的位置。

样例输入

3 10 5

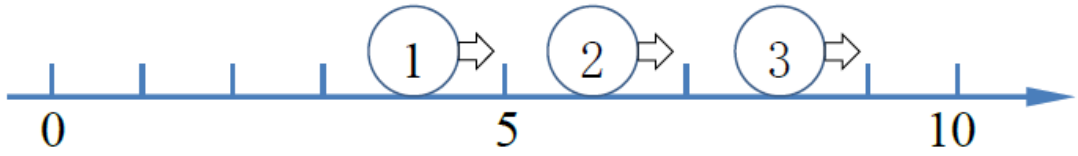
4 6 8

样例输出

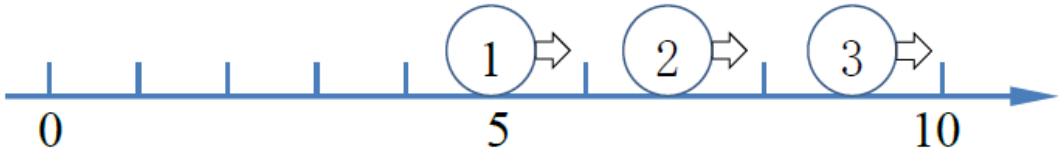
7 9 9

样例说明

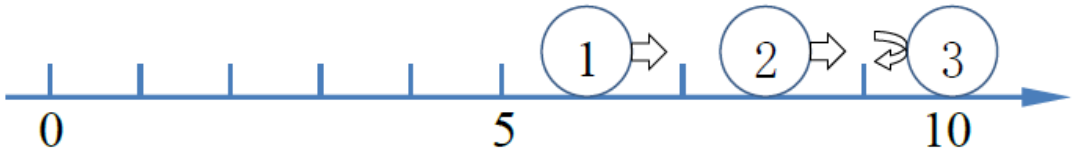
初始时，三个小球的位置分别为 4，6，8。



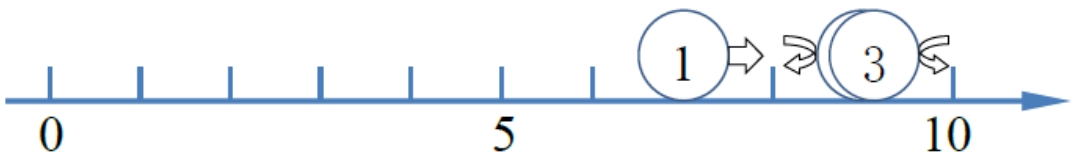
一秒后，三个小球的位置分别为 5，7，9。



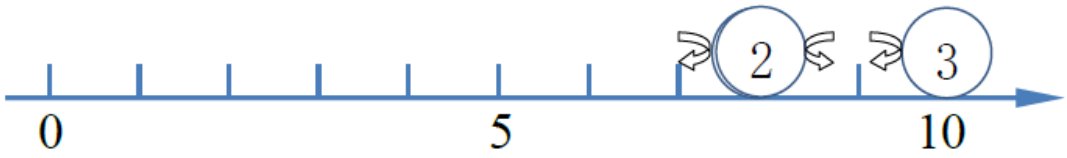
两秒后，第三个小球碰到墙壁，速度反向，三个小球位置分别为 6，8，10。



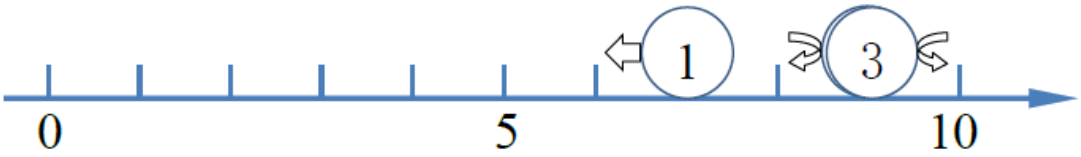
三秒后，第二个小球与第三个小球在位置 9 发生碰撞，速度反向（注意碰撞位置不一定为偶数），三个小球位置分别为 7，9，9。



四秒后，第一个小球与第二个小球在位置 8 发生碰撞，速度反向，第三个小球碰到墙壁，速度反向，三个小球位置分别为 8，8，10。



五秒后，三个小球的位置分别为 7，9，9。



样例输入

10 22 30

14 12 16 6 10 2 8 20 18 4

样例输出

6 6 8 2 4 0 4 12 10 2

问题分析：该问题中，只需维护好每个小球的当前位置，并随时间进行变化，最后按题设要求进行输出即可，本题目实质是一道简单的模拟题，与树、图无关。

选做题：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. 编写拓扑排序算法；

3. 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。

选做题：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 数字排序

问题描述

　　给定 n 个整数，请统计出每个整数出现的次数，按出现次数从多到少的顺序输出。

输入格式

输入的第一行包含一个整数 n，表示给定数字的个数。

第二行包含 n 个整数，相邻的整数之间用一个空格分隔，表示所给定的整数。

输出格式

输出多行，每行包含两个整数，分别表示一个给定的整数和它出现的次数。按出现次数递减的顺序输出。如果两个整数出现的次数一样多，则先输出值较小的，然后输出值较大的。

样例输入

12

5 2 3 3 1 3 4 2 5 2 3 5

样例输出

3 4

2 3

5 3

1 1

4 1

问题分析：该题目可用一个数组，以下标作为数，数组内容存储该数出现次数来实现（这就相当于直接映射，是机试题目里面常用的一种解题法，很多看似非线性的题型最后其实都可以采取哈希或者映射的方法来巧解，尤其是一些看似是树形结构的模拟类题目，要好好体会哈希思想在机试题目中的巧用）。

选做题：习题集 9.32 10.32 10.34

第三阶段总结