

北京大学信息科学技术学院考试试卷

考试科目： 数据结构与算法实习 姓名： _____ 学号： _____

考试时间： 2015 年 1 月 5 日 任课教师： _____

题号	一	二	三	总分
分数				
阅卷人				

装订线内

不要答题

北京大学考场纪律

- 1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。
- 2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机等）不得带入座位，已经带入考场的必须放在监考人员指定的位置，并关闭手机等一切电子设备。
- 3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束监考人员宣布收卷时，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。
- 4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号，不准携带与考试内容相关的材料参加考试，不准抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有严重违纪或作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学习纪律管理规定》及其他相关规定严肃处理。
- 5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。
- 学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 页。

得分

一、简答题（共 30 分）

1. (20 分) 如下程序是求解二元一次方程（广义）的函数（包含错误），请设计一组尽可能少的测试用例覆盖所有的语句，并指出哪些测试用例会触发错误以及说明理由。如果有语句不能被覆盖，请说明理由。（说明：当方程没有根时返回空指针；有根时返回 **root** 指针且 **root** 里保存根的值）

```

1:  double* solve(double* root, double a, double b, double c) {
2:      const double epsilon=0.00000001;
3:      root[0]=root[1]= -1;
4:      if (abs(a) <= epsilon )
5:          if (abs(b) > epsilon) {
6:              root[0] = root[1]=-c / b;
7:              return root;
8:          }
9:          else return NULL;
10:     else root[0]=root[1]= -1;
11:     if (abs(a) > epsilon)
12:         if (abs(b) <= epsilon)
13:             if (-c / a >= 0)
14:                 root[0] = root[1]=sqrt(-c / a);
15:             else return NULL;
16:             else root[0]=root[1]= -1;
17:     else root[0]=root[1]= -1;
18:     if (abs(a) > epsilon)
19:         if (abs(b) > epsilon)
20:             if (b * b - 4 * a * c >= 0) {
21:                 root[0] = (-b + sqrt(b * b - 4 * a * c)) / (2*a);
22:                 root[1] = (-b - sqrt(b * b - 4 * a * c)) / (2*a);
23:             }
24:             else return NULL;
25:             else root[0]=root[1]= -1;
26:     else root[0]=root[1]= -1;
27:     return root;
28: }
```

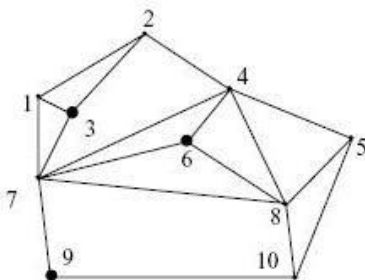
2. (10 分) 阅读以下 C++代码片段，描述该代码片段实现的功能。指出其中出现

的风格不好或错误的代码并将其改正（至少指出五处，相同的问题只算一处）。

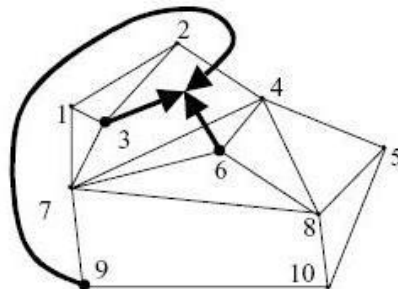
```
1: float num[5] = {4.2, 5.9, 8.34, 6.543, 9.006};
2: float a, b, c;
3: a = num[0];
4: b = num[0];
5: for(int i = 1 ; i <= 5 ; i ++){
6:   if(num[i] >= a)
7:   {
8:     b = a;
9:     a = num[i];
10:  }
11:  else if(num[i]>b) b = num[i];
12: }
13: c = a + b ;
14: cout << c << endl;
```

得分	二、数据结构建模（共 30 分）

1.（10 分）某国修建了很多长城，每段长城都连接且只连接两个城市。不同长城不会相交；两两城市之间最多只有一段长城。该国的任意两个城市都可以通过一段或多段长城连通。这样，整个国土就被这些长城分割成几块空地。从一块空地到另一块，人们必须穿过城市，或者翻越长城（不能在长城上行走）。



(a)



(b)

有一个俱乐部，其成员来自各城市。但是，每个城市最多只能有一人加入该俱乐部，或者一个也没有。

该俱乐部的成员们经常要举行聚会。他们聚会的地点不能在任何一个城市内，也不能在任何一段长城上，而只能在某块空地内。在前往聚会的途中，为了避免交通堵塞，他们不能穿过任何城市；另一方面，由于翻越长城也非一件易事，所以他们也希望尽可能少地翻越长城。为此，他们要选择一块最优的空地来聚会，使得所

有成员到达该空地所需要翻越长城的总次数最少。

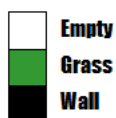
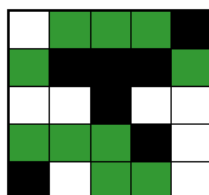
在给定城市 $N([2,250])$ 、空地 $M([2,200])$ 和俱乐部成员 $L([1,3], L \leq N)$ 的家乡等信息后，请建立模型来求解所有成员需要翻越长城的总次数的最小值。

要求：分析中涉及问题抽象、数据结构抽象、算法抽象等过程。不要求写出具体的算法代码。可以采用图 1 为实例，加以说明。

提示：很明显，这个题目整体上需要用图模型。问题抽象，就是分析具体对应到经典图问题中的哪一类问题。数据结构抽象，就是分析在你的模型中，怎样建立顶点，怎样建立边。算法抽象，就是在你建立的图模型基础上，确定采用什么算法思想来求解问题。分析中可以直接应用一些经典的图算法，例如 DFS、BFS、Dijkstra、Floyd、Kruskal、Prim 等；只需要说明你怎么使用经典算法，不需要详细说明这些经典算法的思想。

2. (10 分) 安全安置机器人的问题。

有一个 $N * M (N, M \leq 50)$ 的棋盘，棋盘的每一格是三种类型之一：空地、草地、墙。机器人只能放在空地上。在同一行或同一列的两个机器人，若机器人之间没有墙，则它们可以互相攻击。问给定的棋盘（棋盘规模、空地、草地、墙都给出），求解最多可以安全放置的机器人数目，使得它们之间不能互相攻击。



(1) 请编写一个图模型来表示右图的信息。

(2) 根据你编写的图模型，请说明“安全安置机器人的问题”对应于图论或数据结构中哪一类应用问题。

提示：

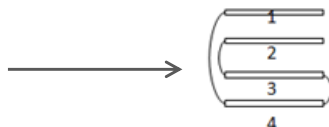
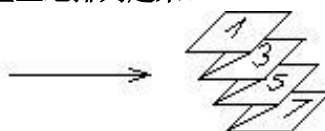
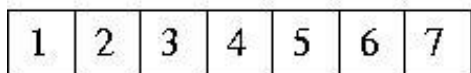
(1) 对问题 1，只需要用结点、边画出一个图，当然需要说明你的结点和边所代表的意义。

(2) 对于问题 2，只需要说明该问题的名字、不需要写出具体的算法。如果你不记得现有的概念，则用自己的语言描述清楚算法目标。

(3) 本题将根据模型及其对应算法的复杂度给成绩，请尽量优化你的模型。

3. (10 分) 有一张 n 个方格子的纸条，格子上无序地写着 1 到 n 这 n 个数字。

每个格子一个数字。现在沿着格子的边线折纸，可以将纸条折成一个小方块（正面只有一个格子）。这样 n 个数字的方格就竖直地排列起来。



输入 n 个连续的数字 ($n \leq 1000$)，代表给定的纸条和纸条上依次写好的数字。请编写算法，判断能否使这些方格上的数字从上到下的排列正好是 1 到 n (无论正反)。例如上图就是两个可行的例子 (7 个格子的纸条和 4 个格子的纸条)。

若能，输出“YES”；否则输出“NO”。(10 分)

要求：

- (1) 算法思路分析过程；
- (2) 描述算法中用到的核心数据结构；
- (3) 带注释的算法框架；
- (4) 算法时间和空间复杂度。

得分

三、数据结构与算法的团队合作考察 (共 40 分)

1. (8 分) 大作业的 Query1 问题可以表述为：查询格式是 `query1(p1, p2, x)`，给出两个人的 ID， $p1$ 和 $p2$ 。再给出一个整数 x 。在子图中找出 $p1$ 到 $p2$ 的最短距离。子图中的边需要满足条件：

条件 1：相连的两个人必须对对方的评论回复超过 x 次

条件 2：两个人有边仅当两个人互相认识 (图为无向图， x 和 y 之间的朋友关系由 $x|y$ 和 $y|x$ 表示)

请回答下面两个问题：

- (1) 请问这个查询在社交网络中有什么现实意义？
- (2) 请描述你们小组解决这个问题所采用的算法，并给出时间复杂度。

2. (8 分) 大作业 Query 2 可表述为：给出一个整数 k 和一个生日日期 d ，找出范围最大的 k 个 interest tag。一个 interest tag 的范围定义为满足以下条件的子图中的最大连通分量：

条件 1：必须有相应的 interest tag

条件 2：必须在日期 d 或者以后出生

条件 3：两个人有边仅当两个人互相认识 (图为无向图， x 和 y 之间的朋友关系由 $x|y$ 和 $y|x$ 表示)

请回答下面两个问题：

- (1) 你们采用的主要数据结构有哪些？
- (2) 请简述算法实现思想。
- (3) 在实现过程中有哪些需要注意的事项？

3. (8 分) 大作业 Query 3 可表述为：给出整数 k ，一个最大距离 h 和一个字符串地点名称 p 。找到相似度最大的 k 个 person 对，最大是指有最多的公共 interest tag。对于每一对 top- k 中的人，他们必须要么住在 p ，要么学习或者工作在 p 。并且，这两个人之间的距离必须不超过 h 。

请回答下面两个问题：

- (1) 你们组解决本题的基本思路是什么？
- (2) 在解决的时候遇到了什么困难？最后是怎么处理的？

4. (8分) 大作业 Query 4 可表述为：给出一个整数 k 和一个字符串 tag 名称 t ，在导出子图中找到 **closeness centrality** 最大的 k 个人。子图满足：

条件 1：其中的每一个人都是某个含有 tag 名称 t 的论坛的成员

条件 2：两个人有边仅当两个人互相认识（图为无向图， x 和 y 之间的朋友关系由 $x|y$ 和 $y|x$ 表示）

点 p 的 **closeness centrality** 的定义： $[(r(p)-1) * (r(p)-1)] / [(n-1) * s(p)]$ ；这里， $r(p)$ 表示由 p 可达点的个数， $s(p)$ 是 p 可达的所有点的距离和。 n 是子图中总的点的个数。

请回答下面两个问题：

(1) 除了你们组最后采用的方法，你们在计算 top- k closeness centrality 的时候还实现过或对比过哪些别的方法？

(2) 你们组在处理这个 query 的时候效率的瓶颈是哪里？

5. (8分) 关于整个大作业

(1) 你们组是怎么组织项目的（哪几个人，任务划分及各位同学的责任）？

(2) 在所有的四个 query 中，你觉得最难做的是哪一个？为什么？

(3) 在设计系统处理这四种 query 时，有没有可能实现过计算结果的重用？比如利用一个 query 的某些计算结果加快另一个 query？（可以是相同类型的 query 也可以是不同类型 query 之间）