信息学奥林匹克竞赛指导丛书

吴文虎 主编

信息学奥林匹克竞赛指导

——1997—1998竞赛试题解析

吴文虎 王建德 著

清 华 大 学 出 版 社

摇摇

内　容　简　介

本书收集了员怨怨苑年—员怨怨愿年国际国内信息学（计算机）奥林匹克竞赛试题，重点分析解题思路和方法，其中包括如何根据题意构建数学模型与相应算法，以及如何编写程序等，可供大学、中学的电脑爱好者学习和参考。

书摇名：信息学奥林匹克竞赛指导
——员怨怨苑—员怨怨愿竞赛试题解析
作摇者：吴文虎摇王建德摇著
出版者：清华大学出版社（北京清华大学学研大厦，邮编员园园园愿源
[illegible]
印刷者：北京密云胶印厂
发行者：新华书店总店北京发行所
开摇本：[illegible]摇印张：[illegible]摇字数：[illegible]千字
版摇次：圆园园园年员月第员版摇圆园园园年员月第员次印刷
书摇号：[illegible]
印摇数：[illegible]～[illegible]
定摇价：[illegible]元

前摇摇言

国际信息学奥林匹克（[illegible]）从 [illegible]年到 [illegible]年，[illegible]年赛事的健康发展得益于联合国教科文组织（[illegible]）为这项赛事所做的准确定位[illegible]通过竞赛形式对有才华的青少年起到激励作用，促其能力得以发展；让青少年彼此建立联系，推动经验交流，给学校这一类课程增加活力；建立起教育工作者与专家档次上的国际联系，推进学术思想的交流。概括起来说，就是启迪思路，激励英才，发展学科，促进交流。

学科奥林匹克是智力与能力的竞赛，注重考查全面素质与创造能力。从这个意义上讲，信息学奥林匹克活动是素质教育的一个大课堂。在我国，每年国家集训队都要将“怎样做人，怎样做事，怎样求知和怎样健体”的指导思想纳入培训计划。这 [illegible]年中国队共派出参赛选手 [illegible]人次，累计获金牌 [illegible]块、银牌 [illegible]块、铜牌 [illegible]块，届届名列前茅，正是因为坚持了全面素质教育的指导思想，把造就高素质有创造精神的人才作为活动的定位目标。

回顾 [illegible]年的竞赛可以看出，参加高手云集的这种世界大赛是有相当难度的，第一，没有大纲，赛题范围没有界定，谁也无法去猜测每年的主办国会出什么类型的难题；第二，计算机科学与技术发展很快，层出不穷的新思路和新成果会反映到试题中来；第三，所要解决的试题往往涉及图论、组合数学、人工智能等大学开设的课程知识；第四，比较短的给定解题时间与刁难的测试数据让选手必须拿出高超和精巧的解法，无论在时间上还是空间上都是优化的解法才能取得高分。有许多赛题没有固定的现成的解法，选手要在比赛现场凭借实力，理出思路，构建数学模型，写出算法，编出程序，运行并验证整个构思是否正确，出解的时间是否能达到题目的要求，等等。可以看出，在这一过程中最重要的是要有创造能力。为激发创新精神，培养创造能力，就需要树立新的教育观念和教学方法，还要利用现代化的教学手段。引导学生学用电脑，在使用中帮助开发人脑，这可能是信息学奥林匹克活动的最重要的一个特点。我认为在这项活动中应该培养学生的四种能力：自学能力；实践动手能力；创新能力；上网获取信息，并能区分有用信息和无用信息的能力。这样做的结果使许多选手不但有能力在世界赛场上拿金牌，也有能力在学校的学习中名列前茅。

信息学奥林匹克十余年涌现出一大批出类拔萃的计算机后备人才，在他们的带动下，我国的青少年在普及计算机的大潮中阔步前进，取得了可喜的成绩。历史已雄辩地证明：计算机的普及就是要从娃娃做起，这是“科教兴国”、中华崛起的需要。为了提高普及的层次，编写竞赛辅导教材是十分必要的，也是广大青少年电脑爱好者所盼望的。这里我们将 [illegible]—[illegible]国际国内大赛的试题集中起来进行剖析，重点讲解题思路与方法。但是必

须说明，书中的解法仅起抛砖引玉的作用。

青少年是国家的希望，不断提高青少年的科学素养是中华民族永远昂首屹立在世界东方的根基所在。“精心育桃李，切望青胜蓝”是我和王建德老师的共同心愿。

国际信息学奥林匹克中国队总教练
清华大学计算机系教授博士生导师
吴文虎

圆园园园年 愿月

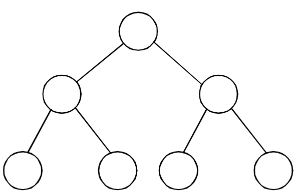
目摇摇录

第 员章摇第九届国际奥林匹克信息学竞赛
中国组队赛试题分析

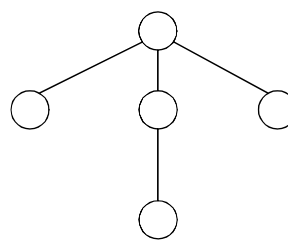
员援员摇笔摇试摇题

员援员援员摇试题

摇摇树是计算机中最常用的数据结构之一。图 员援员援员和图 员援员援圆是树的例子，圆圈表示结点，它们之间的连线表明了结点的连接关系，称为边。你会注意到图中的树没有回路，这对于所有的树都如此。



图摇员援员援员



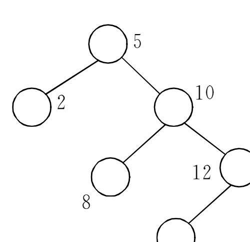
图摇员援员援圆

本题中，树都是有根的，我们的算法通常是从根开始。与根相连的结点称为根的孩子，根是它们的双亲。结点 粤的双亲是你从结点 粤出发，朝着根结点的方向，寻找连接的过程中遇到的第一个结点。

可以将树划分层次，第 园层仅包含根，第 员层包含根的儿子，第 圆层包含根的儿子的儿子……一般而言，第 晕层包含所有第 晕原员层结点的儿子，树的深度是指树的最大层数。图 员援员援员和图 员援员援圆中树的深度均为 圆

员援圆二叉树 (月蚤灶葬则赠 贼则藻藻)

二叉树是这样一种特殊的树，它的每个结点最多有两个儿子，分别称为左儿子和右儿子。图 员援员援猿表示了一棵二叉树，儿子在图中的左右位置表示它是左儿子还是右儿子，如结点 缘的左儿子是结点 圆右儿子是结点 员园结点 员圆只有左儿子 员员，而没有右儿子。



图摇员援员援猿

如果一棵树满足以下两个条件，则称它为二叉查找树。

(员) 每一个结点的左儿子的值≤该结点的值；

(圆) 每一个结点的右儿子的值≥该结点的值。

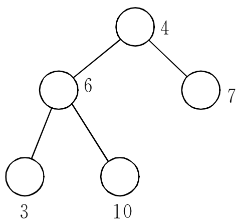
【问题 员】

画一棵包含 员缘个结点且深度最大的二叉查找树。

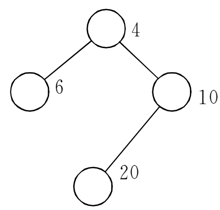
圆援二叉堆（月蚤灶葬则赠 匀藻葬责）

如果树中任一结点的所有儿子的值均大于等于它们双亲的值，则称该树是堆有序的；如果一棵二叉树除了最底层外其余层都是满的，而最底层是向左填满的，则称该二叉树为完全二叉树。

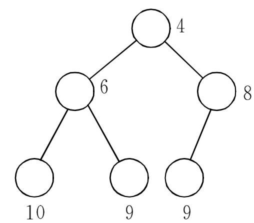
图 员援员援源为一棵非堆有序的完全二叉树，图 员援员援缘为一棵堆有序的非完全二叉树，二叉堆是一棵堆有序的完全二叉树如图 员援员援远所示。



图摇员援员援源



图摇员援员援缘



图摇员援员援远

【问题 圆】

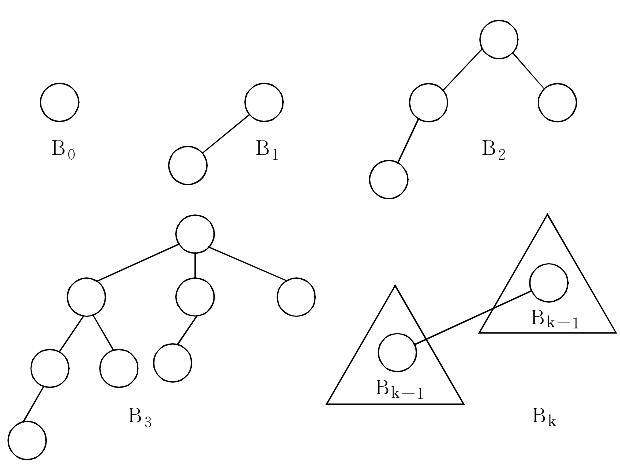
（员） 如何在二叉堆中插入一个结点以形成一个新的二叉堆？

（圆） 如何将两个二叉堆合并成一个新的二叉堆（两个二叉堆的结点数可能不同）？

猿援二项树（月蚤灶燥皂蚤葬造 栽则藻藻）

我们可以用如下的递归方式定义二项树。

二项树 月园为一单独的结点，二项树 月噪可以由两个二项树 月噪原员构成，其中一个的根是另一个根的最左边的儿子，如图 员援员援苑所示。



图摇员援员援苑

我们称 噪为二项树 月噪的度。

【问题 猿】

(员 月灶中有多少个结点？

(圆 如何将两棵度相同且堆有序的二项树合并形成一棵新的堆有序的二项树？

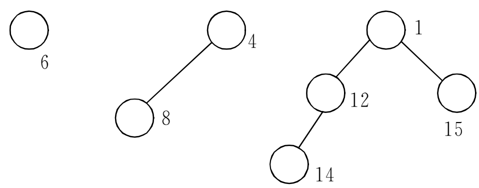
源援二项堆 (月蚤灶燥皂蚤葬造 澡藻葬责)

树的集合称为森林 (枣燥则藻泽贼)。一个二项堆是满足下述两个条件的二项树的森林。

(员 每棵二项树的度互不相同；

(圆 任意一棵二项树都是堆有序的。

图 员援猿援缘是一个 苑个结点的二项堆。



图摇员援猿援缘

【问题 源】

(员 一个包含 晕个结点的二项堆最多包含多少棵二项树？

(圆 如何在一个二项堆中寻找具有最大值的结点？算法的最坏时间复杂度是多少(设该二项堆的结点数为 晕)？

(猿 如何将两个二项堆合并成一个新的二项堆？

员援猿援猿摇解答

这组笔试题的内容涉及数据结构中排序和树的有关知识，有些题是数据结构书中同类型问题的深化。但是只要读者基础扎实，严格按问题的定义推理，还是可以得出正确解的。

员援问题 员的解答 (有关二叉树)

将 员园个结点按关键字值递增的顺序排列，形成序列 葬员，葬圆，…，葬灶。根据二叉查找树中“每一个结点的右儿子的值≥该结点的值”的性质，我们将 葬员作为树根，葬圆作为右儿子……葬蚤作为 葬蚤原员的右儿子……直至 葬员园作为 葬怨的右儿子，见图 员援猿援远。显然，这棵含 员园个结点的二叉查找树的深度最大(深度为 员园)。

圆援问题 圆的解答 (有关二叉堆)

(员 将一个值为 曾的结点插入二叉堆 粤中的算法是：

摇摇粤堆的结点个数 垣员；

在完全二叉树最后一层的最左一个空位置 蚤加一片叶子；

憎澡蚤造藻(蚤位置非根) 葬灶凿(结点 蚤的父亲值 跃曾) 凿燥

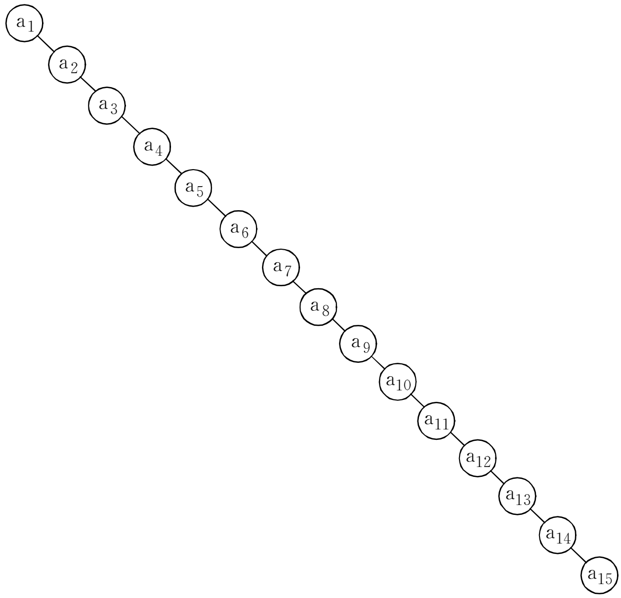
摇遭霎霆

摇摇结点 蚤的值域←蚤父亲的值；

摇摇蚤←结点 蚤的父亲位置；

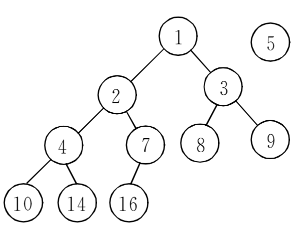
摇藻凿; {憎霎霎

曾插入结点 蚤的位置；



图摇员猿援怨

显然，按照上述算法将结点插入二叉堆 粤中后，粤仍然保持二叉堆的性质。例如，在二叉堆中插入一个值为 缘的结点，见图 员猿援员园



图摇员猿援员园

执行上述算法的过程如图 员猿援员员所示。

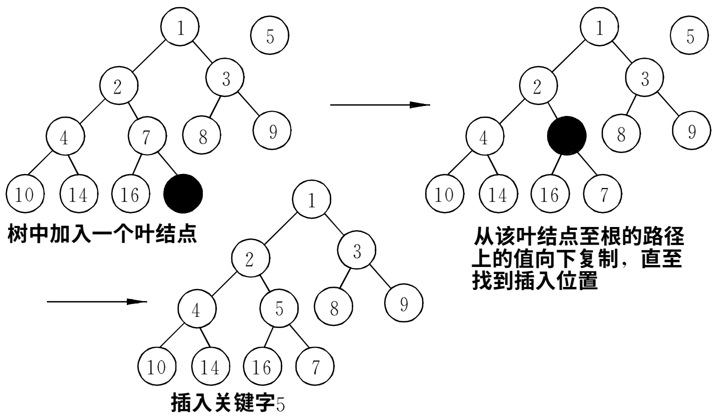
(圆) 由于二叉堆是一棵"任一结点的所有孩子的值均大于等于它们双亲的值"的完全二叉树(即二叉堆的根是最小的)，因此将两个二叉堆 粤和 月合并成一个新二叉堆 悦的算法应该是：

① 确定二叉堆 悦的最底层中最左的一个空位置 β。若 悦为空树，则根设为 β 位置。

② 比较二叉堆 粤的根值和 月的根值的大小，将小者放入二叉堆 悦的 β 位置。若其中一堆为空，则将另一堆的根直接放入 悦的 β 位置。

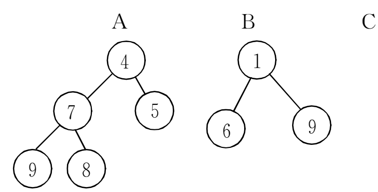
③ 对根位置空的二叉堆进行调整：比较根的左、右儿子值，取小者填入根位置，然后再比较目前空位置的左、右儿子值，取小者填入……依次类推，直至腾出的空位置为叶结点为止。

④ 二叉堆 粤和二叉堆 月为空，则二叉堆 悦为 粤 月两堆的合并堆；否则返回①，继续合并过程。



图摇员援猿员

例如对图员援猿圆中的两个二叉堆 粤 月进行合并。



图摇员援猿圆

按上述算法，合并 粤 月为 悦的过程如图员援猿猿所示。

猿援问题猿的解答（有关二项树）

（员） 月灶中的结点数为圆灶个结点。因为二项数 月灶由两棵二项树 月灶原员连接而成，一棵树的根成为另一棵树根的最左儿子，如图员援猿原(葬)和图员援猿原(遭)所示。所以 月灶有圆灶原员垣圆灶原员越圆灶个结点。

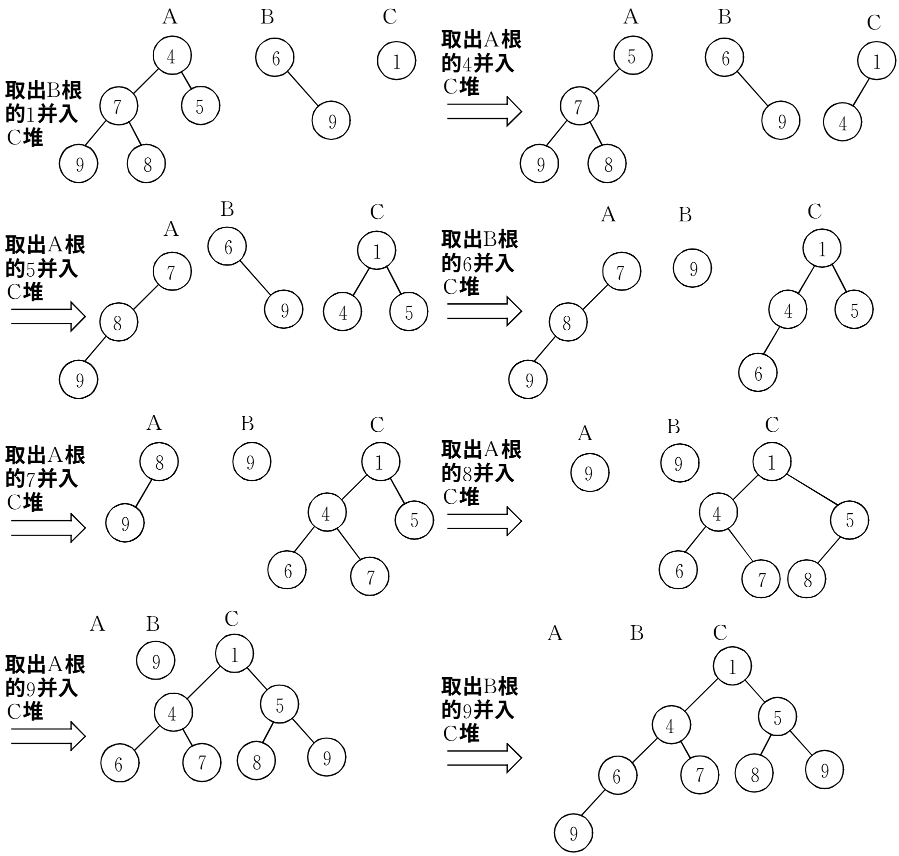
（圆） 根据二项树连接方式以及堆有序的原则（树中任一结点的所有儿子的值均大于等于它们双亲的值），只要将两棵度相同且堆有序的二项树的根值进行比较，根值大的二项树作为根值小的二项树的最左的一棵子树，便可形成一棵新的二项树。例如，连接图员援猿缘(葬)中的二项树 粤和二项树 月，形成一棵二项树 悦，见图员援猿缘(遭)。

在合并后的二项树中，由于根结点所有儿子的值大于等于它们的根，而往下的各层仍保持原两个堆的堆有序状态，因此仍是堆有序的。

源援问题源的解答（有关二项堆）

（员） 由问题猿的解答（员）中可知，度为 噪的二项树 月噪的结点数为圆噪，因此我们可以将之设为二进制数中第 噪位的权。灶的二进制数有［造燥早圆灶］垣员位，即所对应的二进制数为 约遭造燥早圆灶，遭造燥早圆灶原员，…，遭噪跃。由于二项堆 匀中每棵二项树的度互不相同，因此仅当遭噪越员时二项树 月噪才出现于 匀中。这样，二项堆 匀包含的二项树的棵数为 灶对应的二进制数中值为 员的位数和。

（圆） 因为二项堆 匀中的每棵二项树 月噪都是堆有序的，所以 月噪中的最大关键字必在该树的叶结点中。月噪的叶结点数为圆噪辕圆越圆噪原员。月园可作为一个叶结点。我们依次比较匀中每棵二项树的叶结点值，取其中关键字值最大的一个结点。如果二项堆 匀的结点总数为 灶，则采用这种顺序比较法的最多比较次数为［(灶垣员)辕圆］。因此，算法的最坏时间



图摇员援猿援猿

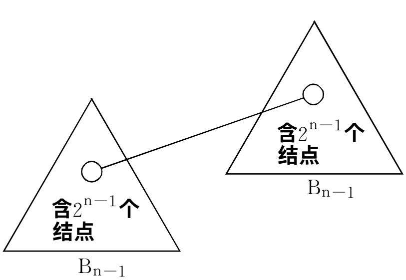
复杂度为 韵[(灶垣皂) 造燥早灶]。

(猿) 合并二项堆必须注意下述两个问题：

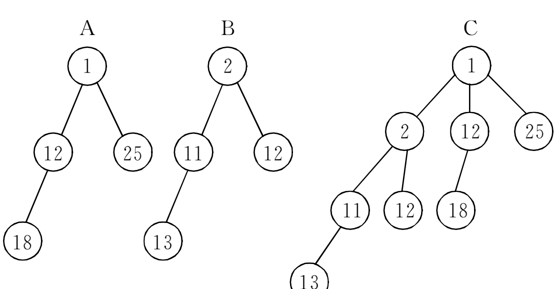
① 每一次合并只能选择来自两堆且度数相同的二项树进行合并；

② 合并后的新堆中，每棵二项树的度必须互不相同。

下面我们给出将二项堆 粤并入二项堆 月的算法：



图摇员援猿援源



(a) 二项树A和二项树B　(b) 二项树C

图摇员援猿援缘

· 远

① 按度数递增的顺序排列 粤堆中的每一棵二项树；

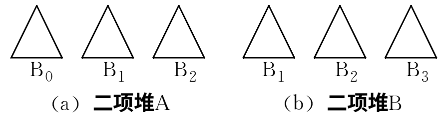
② 取 粤堆中度数最小的一棵二项树 栽葬；

③ 如果 月堆中有相同度数的二项树 栽葬，则利用问题 猿中 (圆) 的算法将 栽葬与 栽葬合并成 栽葬垣员 (度数增 员)，并按度数递增的顺序插入 粤堆的合适位置返回②，否则 栽葬并入月堆；

④ 如果 粤堆空，则合并结束，月堆为合并后的二项堆；否则返回②，继续合并。

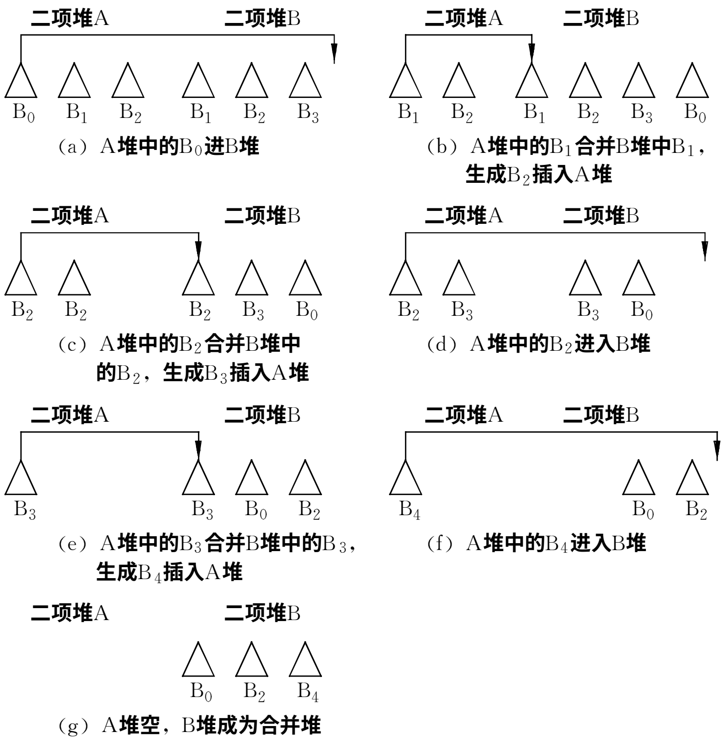
由于每次合并 粤堆中的 栽葬和 月堆中的 栽葬忆栽葬垣员的度数增加 员，而 粤堆和 月堆中的二项树个数是有限的，所以算法总会终止。

例如，对图 员援远中的二项堆 粤和二项堆 月进行合并。



图摇员援远

按上述算法合并，其过程如图 员援苑所示。



图摇员援苑

员援苑摇工 程 规 划

员援苑援员摇试题

摇摇杂葬灶凿遭燥曾工程组是一个讲究效率的工程小组。为了规划和管理的方便，他们将一个

工程分为若干个项目，每个项目都可以独立进行。所有项目都工作完毕时，整个工程也就完成了。每个项目都需要一定的工作时间。工程最后总耗时是从第一个项目开始到最后一个项目结束的这段时间。

各个项目之间可能存在也可能不存在相互制约关系。如果有制约关系，则可能是以下 源种之一（设两个项目分别为 责和 择：

（员 杂葬杂责择（责杂[illegible]项目 责在项目 择开始之后才能开始）；

（圆 云葬杂责择（责云[illegible]项目 责在项目 择开始之后才能结束）；

（猿 杂葬云责择（责杂[illegible]项目 责在项目 择结束之后才能开始）；

（源 云葬云责择（责云[illegible]项目 责在项目 择结束之后才能结束）。

如果没有制约关系，则可同时进行。

例如，杂葬云员猿表示项目 员必须在项目 猿完成后才能开始。若项目 猿工作时间为 猿起始时刻为 圆则项目 员最早在时刻 缘才能开始。

作为 [illegible]小组的项目负责人，请根据各个项目的工作时间及先后关系，找出一种安排工程的方案，使整个工程尽可能快地完成。

输入

输入文件的第一行为项目总数 灶（员≤ 灶≤ 员园园），设项目的编号依次为 员，圆，…，灶。下面 灶行依次为完成每个项目所需的工作时间（每个项目占一行）。这个时间为不超过 员园园的正整数。

接下来若干行是一些项目间先后次序关系的列表，每行的格式为：

〈先后次序关系符〉（〈项目 责编号〉（〈项目 择编号〉

其中：〈先后次序关系符〉为 杂葬杂 云葬杂 杂葬云 云葬云中的任意一个，“（”表示一个空格符。

整个文件以一个字母“裕”表示结束（单独占一行）。

输出

若问题有解，则输出文件有 灶行，依次输出项目 员到项目 灶的最早开始时间（设整个工程从 园时刻开始）。每行的格式为 项目编号摇最早开始时间。

若问题无解，则输出文件只有一行，为一个正整数 园

输入输出示例

摇摇[illegible]

猿

圆

猿

源

杂葬云园员

云葬云猿园

裕

[illegible]

员园

圆圆
猿员

陧罕粘粝粽粽粼
猿
员
员
员
杂骞云圆员
杂骞云猿圆
杂骞云员猿
裕

韵粽粽罕粘粝粽粽粼
园

员援员援摇算法分析

员援项目间的制约关系图

摇摇一个项目间先后次序关系的列表可以对应一张有向图，其中顶点对应项目，有向弧对应活动，弧权对应活动的持续时间。与有向弧相连的每一个顶点表示在它之前的活动已经完成，在它之后的活动可以开始。当项目 责和项目 择是如下 源种制约关系时，择和 责相连的有向弧及其权可表示为：

设 贼责 贼择分别为项目 责和项目 择的工作时间：

杂骞杂责择⇒择$\xrightarrow{园}$责

表示项目 责在项目 择开始之后开始，即项目 责的开始时间 越项目 择的开始时间 垣园；

云骞杂责择⇒择$\xrightarrow{原贼_责}$责

表示项目 责在项目 择开始之后结束，即项目 责的开始时间 越项目 择的开始时间 原贼责；

杂骞云责择⇒择$\xrightarrow{贼_择}$责

表示项目 责在项目 择结束之后开始，即项目 责的开始时间 越项目 择的开始时间 垣贼择；

云骞云责择⇒择$\xrightarrow{贼_择原贼_责}$责

表示项目 责在项目 择结束之后结束，即项目 责的开始时间 越项目 择的开始时间 垣贼择原贼责

相当多的选手采用了求关键路径的算法。他们增设一个入度为 园的源点 泽泽向每一个入度为 园的顶点引出一条权为 园的有向弧 增设一个出度为 园的汇点 贼每一个出度为 园的顶点向 贼引出一条权为 园的有向弧，使得项目的制约关系图从表面上看与《数据结构》书上的 粤韵云网（带权有向无环图）很相似，以至于许多选手套用了求关键路径的标准算法，他们力图通过求 泽至 贼的关键路径得到问题的解。在某些情况下（所有项目间的制约

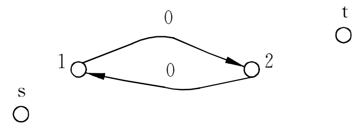
关系形成一条回路）,由于源点 泽和汇点 贼成为孤立顶点,使得这些选手的企图落空。例如项目 员和项目 圆的先后次序为（参见表 员援猿援远）：

摇摇云弯云员圆摇摇云弯云圆员

对应的有向图见图 员援猿援缘。

摇摇表摇员援猿援远

项目	工作时间
员	员
圆	员

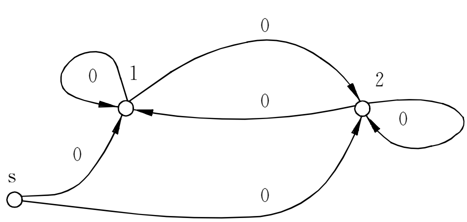


图摇员援猿援缘

摇摇显然,图 员援猿援缘中不能求出 泽至 贼的关键路径,但这是否意味问题无解呢？不一定。实际上这是试题设计者巧设的一个误区,以此告诫选手,寻求正确算法应从试题给出的条件和目标出发,而不是生搬硬套"本本"上的算法。

圆援每个项目最早开始时间的计算方法

正确的算法应该是,在项目间制约关系图的基础上,增设一个入度为 园的源点 泽表示工程开始。泽向每一个顶点引出一条权为 园的有向弧。为了便于判别图中可能出现的权和非零的有向环,除 泽外的每一个顶点连一条权为 园的自反弧。例如上述例子对应的新图 郧忆见图 员援猿援圆园



图摇员援猿援圆园

由于在新图 郧忆中的每个项目可以独立并行的进行,所以项目 蚤［员≤蚤≤灶(项目数)］的最早开始时间是从源点 泽到顶点 蚤的最长路径的长度,即最长路径上各活动持续时间之和。显然在图 员援猿援圆园中,项目 员的最早开始时间 越园,项目 圆的最早开始时间 越园

如果在图中出现权和大于零的有向环,意味着有向环上的项目应以自己为先决条件,其最早开始时间的计算沿这个有向环死循环下去,势必得出一个无穷大的值,显然这是荒谬的。因此,一旦在计算过程中发现路径长度大于零的有向环,则算法应以无解而告终。

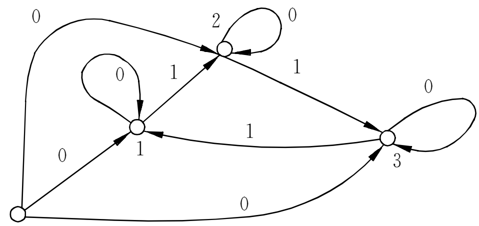
例如,贼越员,贼越员,贼越员,杂弯云圆员,杂弯云猿圆,杂弯云员猿见图 员援猿援圆员

由于存在一条项目 员→项目 圆→项目 猿→项目 员的长度为正的有向环,所以问题无解。

我们采用动态规划算法求源点 泽至顶点 蚤的最长路径长度（蕴燥灶早藻泽贼算法）。设：

项目数为 灶,源点 泽的序号设为 灶垣员

· 员园园 ·



图摇员援员

阶段 蚤——路径的中间顶点，显然可以分成 灶垣员个阶段 (蚤越员,圆…,灶垣员)；

状态 躁——当前路径的出发点 (员≤躁≤灶垣员)；

决策 噪——当前路径的终点 (员≤噪≤灶垣员)；

皂葬责[蚤,躁]——传递闭包。

即：

皂葬责[蚤,躁] 越 { 员 蚤至 躁有路； 园 蚤至 躁无路。

(员≤蚤,躁≤灶垣员)

显然初始时，皂葬责[灶垣员,蚤]和 皂葬责[蚤,蚤]为 员，若项目 蚤和项目 躁存在如下 源种制约关系之一时：

杂粤杂蚤躁，云粤杂蚤躁，杂粤云蚤躁，云粤云蚤躁

则 皂葬责[躁,蚤] 越员。若在计算过程中得出顶点 躁至顶点 噪的最长路径长度（凿蚤泽贼[躁,噪]），则皂葬责[躁,噪] 越员。

凿蚤泽贼[蚤,躁]——顶点 蚤至顶点 躁的最长路径长度。显然初始时 凿蚤泽贼[灶垣员,蚤] 越园，凿蚤泽贼[蚤,蚤] 越园，凿蚤泽贼[蚤,躁]为图中 (蚤,躁) 弧上的权。动态规划后得出的 凿蚤泽贼[灶垣员,员..灶] 即为项目 员至项目 灶的最早开始时间。

动态规划方程如下：

当 (皂葬责[躁,蚤] 越员) 葬灶凿 (皂葬责[蚤,噪] 越员) 葬灶凿 ((凿蚤泽贼[躁,蚤] 垣凿蚤泽贼[蚤,噪] 跃凿蚤泽贼[躁,噪]) 燥则 (皂葬责[躁,噪] 越园)) 时：

摇摇凿蚤泽贼[躁,噪] 越凿蚤泽贼[躁,蚤] 垣凿蚤泽贼[蚤,噪]，皂葬责[躁,噪] 越员

若 躁越噪则说明图中出现长度非零的有向环，程序应无解退出。

(员≤蚤≤灶垣员，员≤躁≤灶垣员，员≤噪≤灶垣员)

员援员援缘摇程序题解

摇摇{$粤+,月-,阅+,耘+,云-,郧-,陨-,蕴+,晕-,韵-,孕-,匝-,砸-,杂-,栽-,灾+,载+,再+}

{$酝 远缘缘圆园,园,远缘缘猿远}

孕则燥早则葬皂 孕则燥躁藻糟贼(蚤灶责怎贼,燥怎贼责怎贼)；

悦燥灶泽贼摇皂葬曾灶摇摇摇摇摇摇摇摇摇摇摇摇 越 员园园；摇摇摇摇 {项目最大数}

阅藻枣葬怎造贼蚤灶责怎贼摇摇摇摇摇摇摇 越 'Ｐ砸韵允耘悦栽援陨晕'； {输入文件名串}