

文章编号: 1672-5913(2011)16-0052-05

中图分类号: G642

文献标识码: A

ACM/ICPC 亚洲区预选赛命题经验谈

郭 炜

(北京大学 信息科学技术学院, 北京 100871)

摘 要: ACM 国际大学生程序设计竞赛, 是世界上规模和影响力最大的大学生程序设计竞赛。从 2008 年至 2010 年, 笔者依托北京大学, 先后承担了国内 4 次亚洲区预选赛的命题工作, 积累了一定经验, 也有一些教训, 写出来与大家分享, 希望对兄弟院校今后的命题工作, 以及 ACM/ICPC 选手和教练们有参考价值。

关键词: ACM/ICPC; 亚洲区预选赛; 题目; 数据; 算法

1 ACM/ICPC 的题目形式、赛制及中国区命题的现状

ACM 国际大学生程序设计竞赛(ACM International Collegiate Programming Contest, 简称 ACM/ICPC)是全球规模最大, 最有影响力的大学生程序设计竞赛, 它始于 1970 年, 到 2010 年为止已经举办了 35 届。

ACM/ICPC 的形式是三名队员在 5 小时内用一台电脑编程解决 8~12 道题。每道题由英文题目描述(题面)、标准输入数据和标准输出数据三部分组成。英文题面中包含很小一部分输入数据及其对应的输出数据, 以作为 Sample(样例)。参赛者须根据题目描述编写程序, 该程序读入标准输入数据, 其输出必须和标准输出数据完全一致, 而且要在题目规定的时限内(通常是两秒, 超过 10 秒的极少)算出结果, 才算通过该题。完整的标准输入数据和标准输出数据选手们是看不到的, 选手的程序提交到裁判的机器上, 裁判以标准输入数据作为输入运行其程序, 然后取得该程序输出结果和标准输出数据进行比对, 以判断该程序是否通过。必要时也会用人工来比对输出结果。比赛最终以通过题目的数量多少来决定名次, 过题数相同的, 以做题速度区分高下。

例如, 著名的“A+B Problem”, 题目就是: 输入两个数, 请输出他们的和。标准输入数据是成对的数, 标准输出数据是每对数的和。当然, 真实的竞赛是不会有这么简单的题目, 这里只是用它来说明题目的形式而已。

ACM/ICPC 的竞赛题, 都是需要编程解决的问题。

因此, 一道完整的题目, 除了英文题面、输入和输出数据外, 命题者还要提供标准程序(标程)。标程必须在题面规定的时限内, 由标准输入数据计算出标准输出数据。

ACM/ICPC 的赛制是每年的 9 月到 12 月在各大洲进行预选赛, 选拔出优秀队伍 100 支左右参加第二年 3、4 月份进行的全球总决赛。

目前, 在预选赛阶段, 中国大陆有 5 个赛站(赛站并不固定)。因此, 每年有 5 个大学会承办 ACM/ICPC 亚洲区预选赛。例如, 2010 年, 哈尔滨工程大学承办哈尔滨赛站的比赛, 天津大学承办天津赛站的比赛, 浙江理工大学承办杭州赛站的比赛, 四川大学承办成都赛站比赛, 福州大学承办福州赛站比赛。

中国的大学申办 ACM/ICPC 亚洲区预选赛比较踊跃。但是, 有信心或愿意独自命题的学校并不多(为何如此后文会解释)。因此, 许多大学在承办比赛时, 会委托 ACM/ICPC 实力较强的学校代为命题。除北京大学外, 复旦大学、浙江大学、中山大学都曾为别的大学举办的亚洲区预选赛命题。

笔者依托北京大学, 从 2008 年至 2010 年, 先后为国内 4 次亚洲区预选赛命题。这 4 次比赛的承办者分别是: 北京交通大学(2008 年)、浙江大学宁波理工学院(2009 年)、浙江理工大学(2010 年)和福州大学(2010 年)。

表 1 是最近 11 场中国大陆亚洲区预选赛的比赛结果统计, 其中打星号的比赛由北京大学负责命题。

北京大学的命题区分度好、难度适中, 从未 Rejudge (因题目错误, 或其他原因导致必须对选手们

作者简介: 郭炜, 男, 讲师, 北京大学 ACM 国际大学生程序设计竞赛队教练, 研究方向为人工智能。

提交的程序重新评判), 因而获得参赛选手和教练们的广泛好评。2009年ACM/ICPC中国区指导委员会曾经评选过“优秀命题奖”, 北京大学获得此项荣誉。

表 1 最近 11 场中国大陆亚洲区预选赛的比赛结果统计

比赛时间/地点	参赛队数	过0题的队数	总题数	没人过的题数	现场最多过题数	金奖题数	银奖题数	铜奖题数
2008 / 北京*	100	10	10	0	9	N/A	N/A	N/A
2009 / 宁波*	130	16	10	0	8	6-8	4-5	3-4
2009 / 武汉	132	1	10	2	8	6-7	4-5	3-4
2009 / 合肥	102	2	10	2	7	4-6	3-4	3-3
2009 / 上海	135	20	10	2	7	3-7	2-3	1-2
2009 / 哈尔滨	104	0	11	3	7	4-6	3-4	2-3
2010 / 哈尔滨	126	0	10	0	10	7-9	5-7	3-5
2010 / 天津	131	10	10	0	9	6-9	4-6	2-4
2010 / 杭州*	138	6	10	1	8	5-8	4-5	2-4
2010 / 福州*	153	10	10	0	10	7-10	5-7	2-5
2010 / 成都	126	0	11	2	9	6-9	4-6	3-4

注: 1. 其中打*(星号)的比赛是由北京大学负责命题。

2. N/A 笔者未获得数据。

3. 有几场比赛的现场最多过题数, 和金奖最多题数不一致, 是因为在这些比赛中, 过题最多的队伍都是不参加排名的队伍。

2 ACM/ICPC 对题目的要求

笔者从2004年起担任北京大学ACM/ICPC竞赛教练, 率队参加过二十多次亚洲区预选赛和六次总决赛, 此外还有一定命题经验, 因而对什么题目是ACM/ICPC的好题, 有一点自己的看法, 希望能与大家探讨。

笔者认为, 一道ACM/ICPC的好题, 应该满足以下条件:

1) 题面和数据正确。这是最基本的要求。其中, 常见的错误是输入数据的范围超过了题面所说的范围。

2) 题目描述易于理解, 没有歧义。中国人用英

文写题面给中国人看, 如果出题者不仔细斟酌, 不同人读题很可能就会有不同的理解。选手按错误的理解去编程, 就是一场灾难。这样的题目, 必然会遭人诟病。虽然赛场上可以在有人提问后发现二义性问题并且发Clarification予以澄清, 但还是比较狼狈的。

3) 数据要足够强。所谓足够强, 有两个方面的含义, 一是指数据里应包含各类边界条件、特殊情况等(如果有的话), 使得做题者一旦有某个地方考虑不周, 就会得出错误答案; 二是数据应尽可能让那些算法时间复杂度比标程高五六倍的程序不能在题目规定的时限内算出结果。这第二条有相当的难度, 因为, 有些程序的算法, 其平均复杂度并不比标程更高, 只有在最坏情况下才会比标程慢很多, 为了要卡住这样的程序, 设计的数据就要包含各种投机取巧算法的最坏情况, 使得那些算法无法通过检验。

4) 程序需要有一定的代码量。毕竟这是程序设计竞赛, 如果十几行代码就能完成, 就算题目设计精巧, 思路巧妙, 也难说是个好题。当然, 比赛中最简单的题目, 可以不作此要求。

5) 题面不能太短, 而且要有个故事。题目最好不要是纯粹的一个已经抽象好的算法或数学问题, 应该用一个故事将这个问题包装起来, 这样才能对参赛选手的英文阅读能力和问题的抽象能力有一定要求。

6) 题目要有新意。不要和陈题类似, 也不要单纯地、不加变化地考某个算法, 即不要让选手照抄某个算法的模板就能通过, 最好让选手在建模上就需要花费时间思考。

在满足上述几条的基础上, 若还能一题多解, 则更是锦上添花。

由上所述可以看出, ACM/ICPC竞赛题, 尤其是中等以上难度的题目, 对命题者有很高的要求。命题者不但英文要好, 而且还要考虑问题足够缜密; 不但要通晓题目所用到的算法原理, 而且还必须具有编程实现该算法的能力; 当然, 还要有一定的想象力和创造性, 才能编出有新意的题目, 包括题目里的故事也要生动有趣。

ACM/ICPC命题费时费力, 对于没有参赛经验的人来说, 哪怕是专门研究算法的教授, 出一道好题也并非易事, 何况教授们往往也不愿意在此事上花费精力。所以, 到目前为止, 国内亚洲区预选赛的绝大多数题目, 都是由ACM/ICPC的现役或退役选手编写

的。对于ACM/ICPC实力不强的学校来说,没有足够多的高水平选手,为亚洲区预选赛命题就成为一项比较困难的任务。所以,许多承办亚洲区预选赛的大学,往往请别的学校代为命题,或自己出一些相对简单的题目,委托其他学校出难题。

前面说了一道题怎样才算好题,下面再来谈谈一套什么样的题才能算好题。

ACM/ICPC官方提到过,一套好题应满足以下三个条件(当然不要出错是起码的):

- 1) 每个参赛队伍至少能通过一道题。
- 2) 每一道题目都有队伍能够通过。
- 3) 没有队伍能够做出所有的题目。

这三个条件得到了大家的广泛认可。但是以笔者经验看,一套题目,只满足这三个条件,还不足以称为好题。笔者个人认为,一套好题,还应该满足以下条件:

- 4) 比赛结果有足够的区分度。
- 5) 第一名至少能过7道题(一般比赛都是10道题)。
- 6) 题目覆盖知识面要比较广。
- 7) 题面长短搭配合适。
- 8) 题目要求有一定的代码量。

上述的第1条,并不容易做到。因为现在每个赛站参赛队伍多达120支到150支,有的队伍并非是经过网络预选赛选拔出来的,而是主办方为支持当地竞赛发展而邀请的,队伍水平有限。为确保所有队都能过题而出一个像“A+B Problem”这样的题,没有意思,也不是命题者所愿。在中国大陆2009年和2010年共10场亚洲区预选赛中,只有3场做到了所有队都能至少能通过一道题。

第2条也不容易做到。为了防止有参赛队能做出所有的题而使题目显得太简单,或希望通过出难题来显示水平,许多学校会出一两道难度很高,甚至根本不可做的题来压轴。这样的题目,往往导致没有人通过,甚至根本没有人提交。因此,在亚洲区预选赛上,2道题无人能过的现象也不少见,在2009—2010年的10场比赛中,其中5场有2道题无人通过,仅有4场是所有题都有队伍通过。

第3条是普遍现象。在中国大陆赛区,2010年前从来没有一支队能在比赛时通过所有的题目。2010年的哈尔滨赛站,第一次出现了一个队做完所有题目的情况;在后来的福州赛站,则有3支队通过了所有的题。

第4条非常重要,笔者甚至感觉区分度是大多数教练判断一套题目好坏的最重要标准。如果同样的题数的二、三十支队,一半得银奖,一半得铜奖,那么落后的队伍必定心中不服,认为该场比赛偶然性太大。

第5条指的是题目不可太难。国内的比赛曾经不止一次出现过冠军只做了4道题的情况,这样的题目,必然区分度也不好,是很难得到大家好评的。

第6条,一套题目一般应该覆盖到以下算法类型:搜索、计算几何、图论、数学、动态规划、数据结构、模拟。

第7条,因为总决赛题面一般较长,英文阅读能力对于总决赛上取得好成绩也比较重要,因此笔者认为短题目不应太多。

第8条,毕竟是程序设计竞赛,不能光靠思考能力取胜,代码编写能力也很重要。因此题目应该要求有一定的代码量,这也和总决赛的要求吻合。

3 北京大学命题的指导思想

笔者一般按照以下几条指导思想进行命题:

1) 题目的难度分布一般是2道最简单的题,2道中等偏易题,3道中等难度题,3道难题。希望一个半小时内能有队伍通过4题。

2) 比赛时不可能做出的题不出。

3) 最容易的题目应该是开赛10到15分钟即被做出,但很弱的队,也要等到最后一小时才能将其通过,这样的题才有技术含量。

4) 倾向于让选手们多过题,希望第一名能通过9题。选手们辛辛苦苦训练一年,为的就是体验在赛场上过题的感觉,那么命题者就应尽量让选手们多体验一下这种感觉。因此,题目不可太难。但是北京大学近来4次的命题,总有10支左右的队伍一道题也做不出,这是否和让选手们多过题的思想矛盾呢?其实不然。因为在我校出的题目中,最简单的题目一般都是用多重循环穷举即可解决的,这样的题目如果5个小时都做不出来,只能说明做题的队伍根本没有经过任何训练。没有付出,自然也应该没有收获。

5) 计算几何、搜索、动态规划是必考的。计算几何还常考两道,其他题型基本上也都能覆盖到。

6) 题面一般较长,向总决赛靠拢。笔者倾向于去掉Sample Input(输入样例)和Sample Output(输出样例)的部分就应该至少占满一页纸。

7) 强调代码实现能力。

4 北京大学命题的过程

从2008年至今,所有题目都由北京大学在校的ACM/ICPC现役/退役队员,以及笔者编写。共有10名队员参加过ACM/ICPC亚洲区预选赛的命题,他们全部获得过亚洲区预选赛金奖,其中8名队员参加过总决赛。每套题目一般有6~7名队员参与命题。出题和验题的是同一班人马。每道题由两个人验,每个人出一道题,就要验两道题。出题者须写承诺书,承诺不得向任何命题组以外的人提及有关题目的任何信息。

有的学校采取的是一组人出题,全套出完后另一组人验题的方式。笔者担心,如果验题时才发现难度不合适需要返工,那代价就太大了,因此没有采用这种方式。

笔者的工作是组织学生反复开会讨论题目和验题(一套题一般要开5~6次会),把握难度和题型,修改甚至重写所有题目描述,以及最后检查所有题目的正确性。笔者还完整编写过3道题,其中两道是简单题,一道中等偏易。此外,有3道计算几何题的问题也是笔者提出。具体的流程如下:

- 1) 确定要参加命题的同学以及题目的类型(即要考查的算法)分布和难度分布。

- 2) 根据每个同学的长处,以“自愿+指定”的方式,为同学们分配题目类型和难度。比较意外的是,同学们并没有抢着要出简单题(所有题目给的报酬都是一样的),因为笔者要求简单题也必须有点新意,要有一个好故事,而想一个好的简单题也不容易。同学们同样没有回避出难题,因为能出一道难题,出题者自己也挺有满足感。总之,在出题和验题方面,都没有发生过挑肥拣瘦的事情。

- 3) 等有了题目想法后,同学们进行讨论。根据大家的评价来判断题目的难度是否合适。

- 4) 写出完整的题目并进行验题。

验题时基本上笔者都在场,根据验题者的表现重新评估题目难度。偶尔会发生需要增减难度的情况,一般都可以通过修改题目条件完成。

学生的英文普遍不够好,写出来的题面往往晦涩难懂,歧义较多。因此,修改甚至重写每一道题的题面,是笔者在命题中最花时间的的工作。第一次负责命题时,笔者甚至花了5个小时改写一道很长的模拟题。随着学生年级增长和经验增加,他们写的题面也越来越好,修改题面的工作就会轻松一些。

题面写完后,会找一个有计算机背景的人读一

遍,询问其是否理解了每个细节。如果理解有偏差,就要改题面。与其强调读者再认真再仔细点就不会看错,还不如写题的写得再直白再明确一些。题面改写后,还会返回给出题者看。这一点非常重要,因为对个别题目,笔者可能会忽略了一些细节,导致题目改写后和出题者原意有出入。

笔者认为题面应以容易理解、没有歧义为第一要点,至于英文是否地道,并不重要。英语已是世界通用的语言,没有标准,每个国家都可以有自己的英语。在中国举办,参赛者又几乎全是中国人的比赛,题目带有一点中式英语的风格,理所应当。国内亚洲区预选赛上曾经发生过出题学校为使题目看起来地道而找外国人或英语专家修饰润色,结果在赛场上反而被选手们诟病歧义太多的情况。

在北京大学负责命题的赛站,笔者均带一名命题组的同学到比赛现场参与裁判工作。笔者发现选手们很少就题意提问,而且也从未因题目歧义发布过澄清(Clarification)。这说明我校的题面质量是不错的。

- 5) 笔者汇集所有的题目,包括数据、标程和验题程序,仔细核对每一道题的题面、程序和数据的一致性。核对无误后就可以交付比赛主办方了。

5 命题注意事项

经历过通过多次命题后,笔者总结了以下13条命题注意事项,供大家参考。

- 1) 最难的两三道题应难度基本相同,不能有一道明显最难。即便是一道可做的题目,只要它是明显最难,结果很可能就是没有人能通过,甚至根本没有人做。

- 2) 再难想的题目,都可能有人做出来;只有代码量大、难写的题目,才可能没有人通过。

- 3) 大多数情况下,题目的实际难度都比命题者判断的难度要高。尤其是计算几何题,即便实际上不难的,由于大家对计算几何的恐惧心理,不愿意去做,通过率也会低于相同难度的其他类型题目。

- 4) 不要出需Special Judge(答案不唯一,要特殊评判)的题目,因为用来评判选手输出数据是否正确的那个Special Judge程序,其自身正确性往往难以验证。

- 5) 题目时限应尽可能短,大部分题目应该时限都是3秒以内。这样可以加快裁判速度。尤其是简单题,不要设10秒这样的长时限,否则由于提交次数多,裁判时间长,会导致许多选手因不能及时得到裁判结果而抱怨。

6) 题目的输入输出数据不要太大, 10M以内为宜, 大多数题目应2M以内。简单题或有陷阱的易错题, 由于提交次数多, 更不能有体积大的输入输出数据。因为在比赛的判题系统PC2中, 标准输入输出数据都是在服务器上存放, 裁判的机器每次判题, 都需要从服务器将数据传输到本机, 如果题目数据量过大, 可能导致网络拥塞, 甚至系统崩溃。

7) 输入数据应该有明确的结束方式(比如以两个0结尾, 或数据开头就指明test case的数目), 不要让程序读到EOF了才能判断数据已经结束。

8) 出题者有时会为了看数据方便, 在数据里多个case之间加空行, 最后要注意去掉。

9) 对于每道题, 出题者都要专门写一个验证数据范围的程序, 输出每个变量的取值范围, 看是否和题面描述吻合(不能超范围, 也不能远小于题目所说范围)。此外还要看会不会出现无意义的数字, 如0的0次方、除数为0等。

10) 由于可能改了题面或数据而忘了改数据样例(Sample), 甚至文字编辑软件有可能自动将Sample里的某些字母变成大写, 所以交付前一定要专门将Sample Input和Sample Output从题面里拷贝出来, 分别存为两个文件, 然后以Sample Input文件作为输入运行标程, 生成输出结果再比对Sample Output文件。比对不能用眼睛看, 一定要用文件比较程序。

11) 要检查题面中Sample Input和Sample Output里各个数据出现的次序是否和题目描述的一致, Sample Input数据有没有超范围。

12) 题面由别人改过后, 一定要让出题者再看一遍。

13) 在最终发布之前, 要把上述9至11条全部重新做一遍, 尤其是第10条, 很容易被忽略。

6 命题中的教训

笔者的命题经历中, 也有一些教训:

1) 对于公式之类容易敲错的内容, 不能仅靠出题者自查, 一定要有其他人检查。

第一次命题是在2008年北京交通大学举办的亚洲区预选赛上。该套题目的I题是个积分题, 题目中写了几个公式。虽然笔者一再和出题者强调注意公式不要写错, 但还是有一个公式里错了一个数字。由于笔者不懂那些公式, 所以也没发现。幸而该题是个难题, 只有15支队伍提交, 其中4支通过; 而且要过此题也不是一定要用该公式(或自己也可以推导并看出该公式有误), 所以现场并无队伍提出异议, 该套题目还是受到了广泛的好评。赛后上海交通大学一支为此题所困的队伍发现了这个错误, 并在北京大学在线判题系统(POJ)的BBS上指出来, 我们也承认并且致歉。

2) 同一学校的队员, 知识结构相似, 甚至可能会有共同的知识盲点。今后在竞赛训练中, 和竞赛命题中评估题目难度时, 都应该充分注意这一点。

2010年在福州大学举办的亚洲区预选赛, 是由北京大学命题的。比赛结果是上海交通大学有三支队伍都通过了所有的题目, 说明此套题目难度不足。造成这一结果的其中一个原因, 是一道我们估计只有几支队伍能过的图论“难题”, 事实上成为有近三分之一队伍通过的中等偏易题。赛前福州大学教练吴英杰老师曾对此题归类为“难题”提出过异议, 可惜笔者没有足够重视。因为当初命题组的同学, 除了出题者, 都不知道此题怎么做。事实上, 我校有一支强队在福州不排名参加了该场比赛, 他们也没有做出此题; 而该队里有两名队员, 获得过2010年哈尔滨赛站的冠军和成都赛站的亚军, 实力非常强。

Writing Problems for ACM/ICPC Asia Regional Contests

GUO Wei

(School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China)

Abstract: ACM/ICPC (ACM International Collegiate Programming Contest) is the largest and the most influential programming contest for college students. Since 2008 to 2010, counting on Peking University, I have been in charge of problems writing for 4 Asia regional contests in China. I wrote this article to share my experience and lessons with others. Hope it would be useful for ACM/ICPC contest hosts, contestants and coaches.

Key words: ACM/ICPC; problems; Asia regional contest; data; algorithm

(编辑: 彭远红) 