



北京大学

本科生毕业论文

题目： 程序设计实现能力评价系统
的改进研究及用户行为跟踪

姓 名： 林舒

学 号： 00948159

院 系： 信息科学技术学院

专 业： 计算机科学与技术

指导教师： 李文新

二〇一三年五月

北京大学本科毕业论文导师评阅表

学生姓名	林舒	学生学号	00948159	论文成绩	
学院(系)	信息科学技术学院			学生所在专业	计算机科学与技术
导师姓名	李文新	导师单位/ 所在研究所	计算机科学技术系 人工智能实验室	导师职称	教授
论文题目 (中、英文)	程序设计实现能力评价系统的改进研究及用户行为跟踪 A Research on Improving the Evaluation System for Reflecting Users' Programming Ability and User Behavior Tracking				
导师评语					
(包含对论文的性质、难度、分量、综合训练等是否符合培养目标的目的等评价)					
<p>林舒同学的论文主要研究的是如何改进现有的北京大学在线程序评测系统(POJ),使之更好地服务于编程学习者。作者在论文中分析了现有系统存在的问题:题目利用率不高,自主学习者缺乏指导等,并针对这些问题提出了一套解决方案,即程序员分级系统。在该系统中,用户可以获得适合自己的题目推荐,也可参加为自己量身定制的能力测试。完成这个系统需要对系统中的几千道竞赛题目进行人工分类,工作量是相当大的,同时也需要研究者具有相当高的算法分析设计能力。林舒同学在这项工作中表现出了极强的水平和能力,也能够使用流畅的中英文撰写科技论文。本科期间有一篇国际会议论文被接受。论文工作表明林舒同学具有很好的从事科学研究的潜力。</p>					
导师签名:					
年 月 日					

摘要

随着计算机的普及，计算机程序设计实现能力已经成为大学生们的必备技能。现有的在线程序评测系统不胜枚举，为用户提供了稳定高效的自动程序评测服务，但对于缺乏正确指导的学习者来说，它们对其能力的提升似乎没有明显的帮助，导致他们在学习中感到迷茫。

本文从该问题出发，在现有的“百练”系统上进行改进，设计并实现了一套新系统，分别从三个角度更准确地评价学习者当前的程序设计实现能力，并根据评价结果向学习者推荐合适的任务。此外，新系统能够根据题库中题目难度和类型自动为学习者生成一套有针对性的试题，提供自我测试的途径。

在最后，本文通过对学习者的反馈以及各种行为记录的收集和分析，得到了新系统实际效果的客观评估结果。

关键词：在线评测系统，能力评价，学习指导，用户行为跟踪

Abstract

Thanks to the widespread use of computers, the basic programming ability is becoming an essential skill for almost all the college students. Although many current online judge systems provide stable and efficient services in automated programs testing, it seems that there is limited help to the learners since few study suggestions is given to them. In this case, learners may feel disoriented during their studies.

This paper aims to solve this problem by implementing a system based on one existing platform - Bailian, which will evaluate the learner's programming skills by combining three scores from different perspectives separately, and then recommend some suitable tasks to them according to the grades of both learners and problems. The system also includes an automatic problem set generator to generate contests for self-testing. The quality of these contests is guaranteed since the grades as well as the categories of problems are considered during generating.

The evaluation of actual effectiveness of the improved system is also provided in this paper. By collecting and analyzing the feedbacks from learners as well as the records of learner's behavior, how much has the system changed in learner's programming learning can be seen clearly.

Keywords: online judge system, skill evaluation, study suggestion, user behavior tracking

目 录

第一章 引言	1
第二章 相关系统研究	4
2.1 在线程序评测系统综述	4
2.2 国内相关系统	5
2.3 国外相关系统	7
2.4 北京大学在线评测系统	11
2.4.1 POJ	11
2.4.2 百练及 OpenJudge	11
2.5 小结	12
第三章 初步改进	14
3.1 改进内容回顾	14
3.2 效果及不足	14
第四章 新系统设计和实现	16
4.1 目标	16
4.2 整体设计	16
4.3 前期准备	18

4.3.1	题目按难度分级	18
4.3.2	题目按类型分类	19
4.4	多维度程序设计实现能力评价	20
4.4.1	段位制	20
4.4.2	积累分	21
4.4.3	挑战分	22
4.4.4	贡献分	24
4.5	个性化任务推荐	26
4.6	题集自动生成	27
4.6.1	个人测试	28
4.6.2	自动安排试题	29
第五章	用户行为跟踪	30
5.1	用户量变化分析	30
5.2	题库利用率分析	30
5.3	用户学习效率分析	31
5.4	题集自动生成系统使用频率分析	31
5.5	系统安全性和稳定性分析	31
第六章	总结与展望	32
	致谢	34
	参考文献	35
	附录 A 作者相关科研成果	37

第一章 引言

近年来，计算机技术已逐渐渗入全世界各个领域，计算机俨然成为了各行各业最需要、也是最高效的工具。作为“指挥”计算机工作的重要手段，程序设计和实现的能力高低，将对能否用好这一工具产生重大影响。在数学物理中，需要用程序处理大数据集或求解复杂的方程式；在社会科学中，需要用程序预测某个事物的发展趋势；在艺术领域，需要用程序设计或制作精彩的动画……因此，程序设计和实现课程已不仅仅是计算机学科学生的专利了，众多理科院系，甚至部分应用方向的文科院系，都纷纷将其作为本科生的必修课；而为了自身发展需要而自学编程者，更是不计其数。

世界上现有的在线程序评测系统数不胜数，各自拥有着自己的特点和针对用户群，呈现出一种百花齐放的态势。而北京大学在线程序评测系统因其稳定性较高，且拥有丰富的题目资源，为全国乃至全世界的学习编程者提供了一个非常优秀的在线学习系统，与教学的结合也愈发紧密。然而，题库中的题目难度和类型参差不齐，一直没有被系统地整理过，仅仅是简单地呈现在用户面前，导致无论是学习者还是教师都难以对题目进行正确地取舍。这既降低了学习者的学习效率，也使得一些优秀的题目“埋没”在茫茫题海之中，造成资源浪费。

从学习者的角度来看，他们希望能够更准确地了解自己目前的能力，并获得学习上的帮助和建议，更希望在做题的过程中享受更多的乐趣；从教师的角度来看，他们一方面想客观全面地了解每名学生的能力，因材施教，另一方面也想要在题海中挖掘合适的题目，充分利用已有的资源。

值得一提的是，去年六月份，在线程序评测系统的管理员账号曾收到过这样

一封邮件：

尊敬的管理员：

您好！

POJ 是一个很好的 ACM 在线题库 + 评测系统，口碑声誉俱佳，我十分喜欢。

在此基础上，我有一个关于 rank 系统的建议：

目前的 rank 系统应该是根据解题数和提交数来排的，比较简约，但是参考价值不大，因为解不同的题所需要的技巧和精力差别也会很大。

我建议不妨引入更有趣的积分系统，就如同11的天梯积分等（只是打个比方），加入诸如题目的难度，题目的热门程度，coder 的活跃程度等因子来进行综合积分。

这样更能提升大家向难题出手的热情！

此外，在 SNS 兴起的当下，POJ 如此众多的用户也是一笔极大的资源，说不定稍加开发就能获得很好的用户黏性和珍贵的用户数据。

以上，祝好！

一只喜欢 POJ 并希望它能越做越好的程序猿

这封邮件充分反映了用户的心声，同时也证明了对现有平台改进研究的必要性和紧迫性。

在去年的本科生科研工作中，曾经对现有平台进行一次较大幅度的改进，通过对题目本身、历史提交记录以及人工标注难度等信息，实现对题目的整理分类工作；并在此基础上，引入“段位制”，更加客观、准确、全面地评价学习者的能力，提供一些参考建议；尝试通过多种措施增加学习者的成就感，从而培养学习者对编程的兴趣，并鼓励学习者为后来人提供帮助；同时，能够根据命题人的需要自动从题库中抽取合适的题目组合成试题集。在该系统正式上线半年多以来，已经吸引了约六千多位用户的光顾和使用，并受到众多学习者及教师的广泛好评。但是，由于研究时间和精力有限，该系统缺乏题目类型数据，导致推荐

给用户的题目容易重复，而题集自动生成系统更是因此而难以运用到实际的教学

中。

有了一次比较成功的改进经验之后，本文尝试了一次更大的改进研究，在保持之前改进的基本思路不变的条件下，加入了题目类型数据，并几乎重新设计和实现了一个新系统，使其更加稳定、安全，且能够为学习者提供一个更好更准确的能力评价、任务推荐以及自我测试的学习环境，同时也为教师提供了一个能够更好地了解学生的辅助教学平台。

本研究的意义在于，实现一个侧重于自主学习/课堂讲授的程序设计和实现能力的学习平台。这是目前世界上众多在线程序评测系统所没有的。

本文第二章列举比较了国内外相关系统的优缺点，尝试从“他山”吸取经验，并借鉴一些成功的范例用于对系统的改进，同时着重介绍了北京大学在线程序评测系统的历史及发展现状，这是本文工作的重要基础；第三章简单回顾了之前的改进工作和效果，并指出其仍然存在的问题；第四章详细说明了新系统的设计和实现；第五章通过对用户行为的跟踪，来分析新系统上线后对学习者的实际影响；最后，第六章为本次研究作了一个总结，并提出了今后可以继续改进的方向。

第二章 相关系统研究

2.1 在线程序评测系统综述

1965 年至今，程序自动评价系统逐渐从简单的自动评分系统，发展到与教学管理系统相结合的程序或脚本；随着互联网技术的兴起，各种形式的在线程序评测系统应运而生。^[1]

一般而言，一个在线程序评测系统会提供给用户一些题集，而用户需要不断地尝试去一一解决。常见的题集主要分为两类：

- 练习：没有时间限制，用户可以随意地在上面解决问题，以达到能够熟练进行程序设计和实现的目的；
- 比赛：用户需要在有限时间内解决问题，这可以锻炼用户在压力下编程的能力。

当用户需要解决一个问题时，他们只要认真阅读系统提供的题目描述，选择一个他/她所喜欢的编程语言并按照题目要求进行程序设计，然后将完成的程序提交给系统。系统会在短短几秒内自动评测程序在指定的数据下是否能够正常运行并得到正确的答案，然后将评测结果及时反馈给用户。^[2]

不同的在线程序评测系统对用户排名的规则略有不同。多数针对大学生的系统都遵循类似国际大学生程序设计竞赛（ACM/ICPC）的竞赛规则：当且仅当用户完全解决一个问题的时候，获得该问题的所有分数（该分数会随着时间的流逝逐渐减少），同时，每次提交错误的程序都会有相应的惩罚。^[3] 另外有一些系

统则是按测试点给分，即用户即使不能完全解决一个问题，也可以得到一部分的分数，这种规则多常见于针对中学生的系统。还有少数一些系统（如 TopCoder）会采用其他更为复杂的排名计算方法。^[4]

这些在线程序评测系统可以按照来源分为两类：

- 国内的在线评测系统；
- 国外的在线评测系统。

还可以按照主要针对的用户群来分类：

- 主要针对大学生；
- 主要针对中学生。

下面将通过典型特例，分析国内外各种类型的在线评测系统的特点，比较各自的优缺点。

2.2 国内相关系统

国内许多大学都建立了自己的在线程序评测系统，其中比较知名的、用户量较大的有浙江大学在线程序评测系统 ZOJ^[5]（图 2.1）、杭州电子科技大学在线程序评测系统 HDOJ^[6]（图 2.2）和哈尔滨工业大学在线程序评测系统 HIT^[7]（图 2.3）等。它们主要针对已经有一定能力的学习者甚至是竞赛选手，虽然题库资源丰富，但由于题库中的题目缺乏分类，初学者难以上手。

作为几乎完全由中学生开发、维护的 Vijos^[8]（图 2.4）则是国内针对中学竞赛的在线程序评测系统的代表。难得的中文题面和经常更新的原创题，为国内的初学者提供了许多便利，题目的分类以及用户评价机制也让用户更方便地找到自己需要的题目。

但是，由于题库中的题目以原创题为主，题目本身的正确性以及题面的可读性难以得到保证，使得题目质量并不理想，题目的难度两极分化严重；虽然引入了用户评价机制，但由于完全由用户来自行评判，没有一个统一的标准，参考价

程序设计实现能力评价系统的改进研究及用户行为跟踪



图 2.1: 浙江大学在线程序评测系统 ZOJ 主页



图 2.2: 杭州电子科技大学在线程序评测系统 HDOJ 主页



图 2.3: 哈尔滨工业大学在线程序评测系统 HIT 主页

值并不高。^[18] 再加上管理者均为中学生，缺乏资金、技术支持，该系统稳定性不足，经常因种种原因处于关闭状态。

2.3 国外相关系统

在国外的在线程序评测系统中，不得不提到的是美国的 USACO^[9]（图 2.5）。这个拥有按照难度精心排列的题目、配套讲义以及详细的解题报告的在线程序评测系统，堪称一本优秀的活教材，也是唯一一个完全面向程序设计初学者的学习平台。

不足的是，它自身的特点导致：需要专业人士花费大量的时间和精力来安排题目。因此，USACO 的题目数量少且相对简单，而且近7年都没有再更新过了，这对于日新月异的计算机学科来说，未免有些过时了。

另一个知名的在线程序评测系统是 ACM-ICPC Live Archive^[10]（图 2.6）。这个唯一可以汇总国际大学生程序设计竞赛所有区域赛和总决赛题目的系统，拥有着最新、最全、质量最好的题集，这是其他任何一个系统所不具有的。同时，该系统在计算用户排名时，会根据通过人数的多少为每道题目分配权值，以鼓励用



图 2.4: Vijos 主页

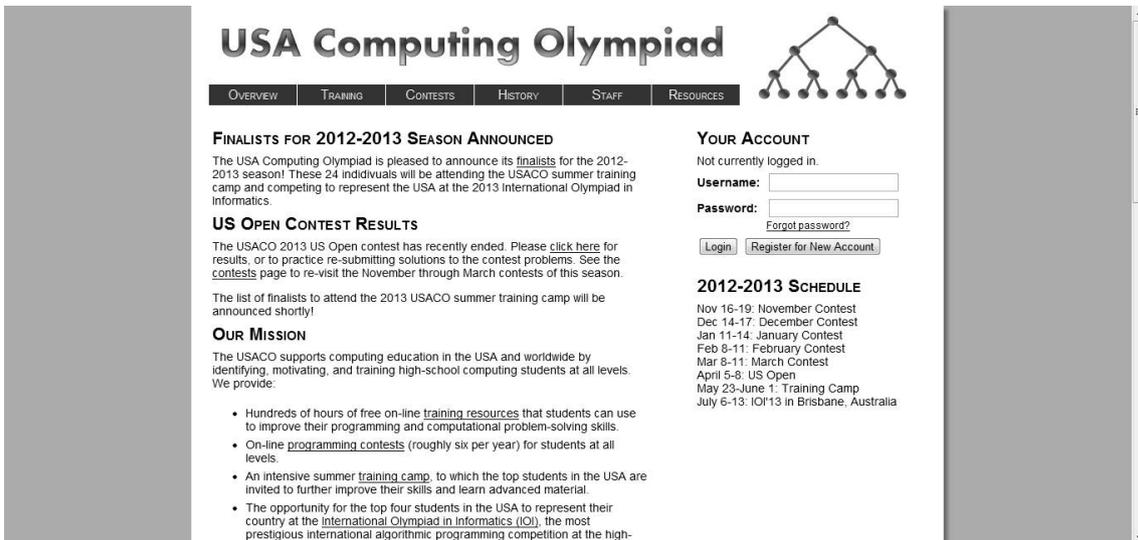


图 2.5: USACO 主页

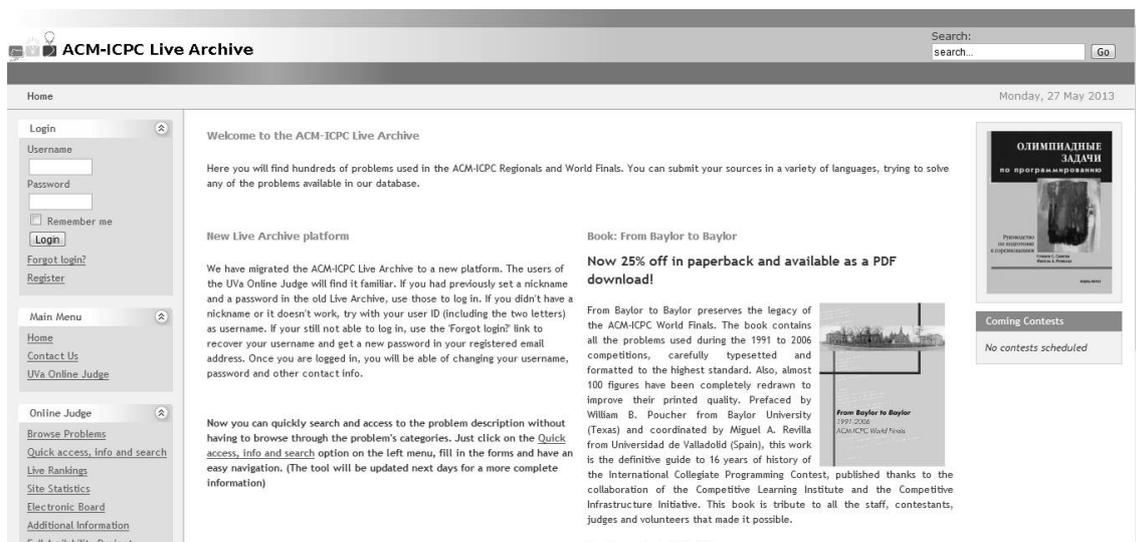


图 2.6: ACM-ICPC Live Archive 主页

户去挑战难题，这是一种比较少见的排名方法。

但它毕竟是 ACM/ICPC 竞赛题目的汇总，题目难度相对较大，不适合初学者。另外，国内用户访问国外网站速度较慢，而且该系统本身不太稳定，需要经常停机维护，系统本身的缺陷给用户体验造成了一定的不良影响。

最后还要提到两个新兴的在线程序评测系统 TopCoder^[11]（图 2.7）和 Codeforces^[12]（图 2.8）。它们均受到了程序设计爱好者的广泛热捧，原因是它们频繁的举办各种级别的比赛，用经常更新的题目、积分甚至是奖金来吸引挑战者。

这两个系统既可以激起参与者的编程热情，同时题目难度有所划分，也方便学习者进行日常训练。然而这种方式成本比较大，需要大量的题目和资金支持，而且大多数用户只关心近期的比赛，之前的题目往往无人问津，题目没有得到充分利用。另外，由于比赛规则的复杂性，对于没有经验的新手来说，上手有一定困难。

其他的一些系统如 SGU^[13]、Timus^[14]等则与国内各大学的系统基本类似。

程序设计实现能力评价系统的改进研究及用户行为跟踪

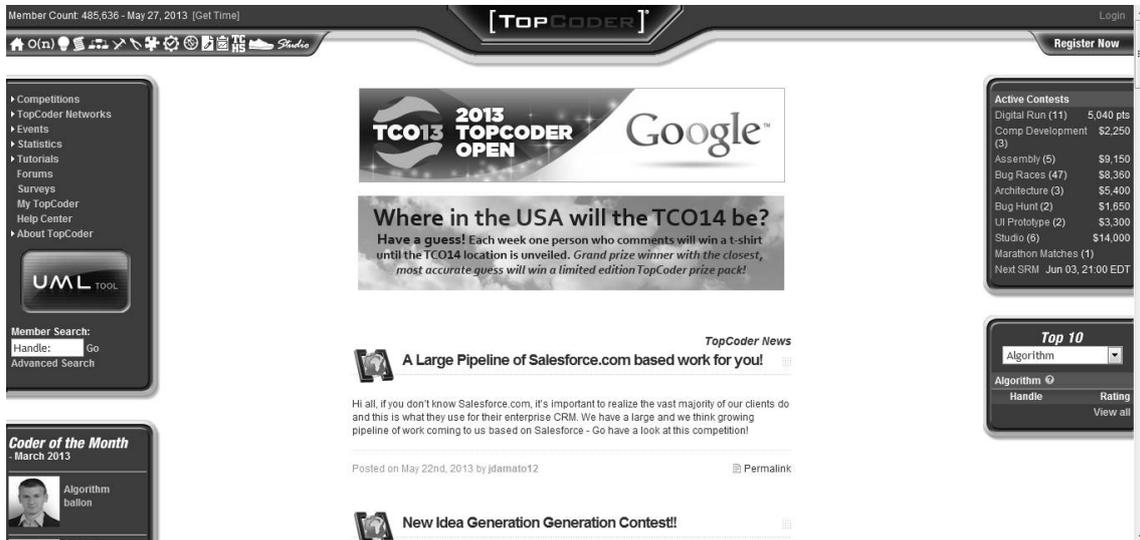


图 2.7: TopCoder 主页

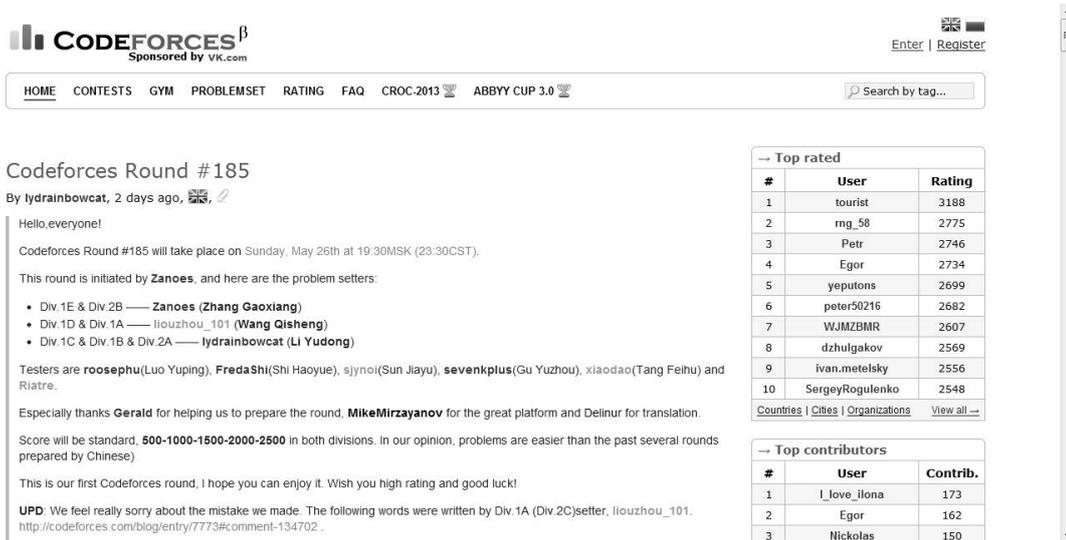


图 2.8: Codeforces 主页

2.4 北京大学在线评测系统

2.4.1 POJ

北京大学在线评测系统 Peking University Online Judge, 简称 POJ^[15]由北京大学人工智能实验室于 2003 年开发上线, 最初网址为 <http://acm.pku.edu.cn/>, 后改为 <http://poj.org/> (图 2.9)。该系统最初的目的是为了本校大学生程序设计竞赛队训练而开发的, 后逐渐被国内甚至国外的编程爱好者熟知, 现在已发展成为集教学和竞赛训练与一身的优秀在线学习系统。^[16]

截止至2013年4月, 该系统在上线不到十年的时间里, 已经积累了 3055 道题目, 拥有超过 50 万的注册用户, 累计程序提交量逾 1000 万。这样的规模, 是国内外同类系统难以比肩的。

该系统代码曾以“Free Version POJ”的形式公开提供给兄弟院校下载, 允许他们用这套代码在自己的服务器上搭建自己的在线评测系统。

POJ 系统与众多国内面向大学生的在线评测系统类似, 大多数题目选自 ACM/ICPC 赛题, 采用标准的 ACM/ICPC 竞赛规则对用户进行排名。^[17]系统题库资源丰富, 但初学者不易上手。

2.4.2 百练及 OpenJudge

北京大学人工智能实验室在 POJ 的基础上, 搭建了一个针对本校学生的相关程序课程教学的全新系统“百炼”(后更名为“百练”)。2010 年, 又进一步开发并上线了开放的在线程序评测平台—OpenJudge (图 2.10), 其目的是为了让兄弟院校能够直接在本平台上建立私有的在线评测系统, 从而有效避免了之前使用“Free Version POJ”配置系统时遇到的各种技术难题, 节约服务器成本。^[19]

迄今为止, OpenJudge 上已创建的小组多达 180 个, 其中不乏中学、大学、甚至一些程序员团体, 而且仍以平均每周一个的速度继续增加着。这足以看出 OpenJudge 的影响力之大。



图 2.9: 北京大学在线评测系统 POJ 主页

OpenJudge 的定位与 POJ 有很大的不同。它更多倾向于针对国内以学习/教学为目的的学生和教师。

2.5 小结

本章介绍了国内外各类型在线程序评测系统的特点。由于大部分系统的设计初衷是为了竞赛训练，而不是基础的程序设计学习，因而不适合初学者自主学习；而少数专门面向学习的系统，又缺乏更新，显得有些过时了。

在专门面向学习和教学这一方面，OpenJudge 已经做了一定的尝试并且有很大的突破，但学习者的学习行为仍然更多地依赖于教师布置作业，且题库利用率不高。我们正是针对这一问题，在该系统上进行一系列的改进，让学习者能够自主地进行学习，同时减轻教师的负担。



图 2.10: OpenJudge 主页

第三章 初步改进

3.1 改进内容回顾

在去年的本科生科研训练中，已经成功实现了对现有系统的一次较大幅度的改进，主要的改进内容如下：

1. 设计并引入“段位制”，根据完成题目数量、参赛成绩以及对系统的贡献等对学习者的程序设计和实现能力进行大致的评估，以作参考；
2. 对题库中将近一千道题目的难度人工标注成六类（按难度从小到大分为1至5级，同时专设有一类“不推荐”）；
3. 从“积累”、“挑战”、“贡献”三个独立的维度对学习者的程序设计和实现能力进行全方位的评价，让学习者更了解自己的真实水平。同时通过一系列激励措施，鼓励学习者去做更有益处的任务；
4. 根据前面得到的能力评估，对每个用户进行个性化的题目推荐，作为给学习者的学习建议，以避免学习者盲目做大大超出自己能力范围的题目、或把宝贵的时间浪费在做大量已经掌握的简单题目上；
5. 能够根据教师的需求自动安排难度适中的试题集，在减轻教师负担的同时，也能够尽可能充分利用现有的题库资源。

3.2 效果及不足

在改进后半年多的时间里，该系统能够正常、稳定地为自学程序设计和实现

的学习者提供了良好的学习指引，为广大学习者所接受。

但是，改进后的系统依然存在较大的问题。其中最大的问题是没有对题目划分类型。

现阶段，我们仅仅是根据难度对题目进行分级而已，但在实际学习过程中，题目的类型也是用户非常关心的属性之一。学习者往往会比较擅长某些类型的题目，而另一些则是他的弱项；举办讲座往往只针对同一种类型的题目，此时若需要在我们的系统中找出一些合适的例题需要花费一定的功夫；生成考题的时候不是单单考虑题目难度就足够的了，还应考虑考察的知识点尽可能不重、不漏……

由于时间和精力所限，之前的改进中未将题目类型划入研究范围，但本次改进的重点，就是将其纳入系统，并对之前的改进内容进行一定的调整，从而进一步完善整个系统，使其发挥更大的作用。

第四章 新系统设计和实现

4.1 目标

我们新系统的最终目标就是，利用现有的题库资源，实现一个让程序设计初学者能够完全自主地学习的平台，同时也能够结合课堂教学，以减轻教师的负担。而我们之后的所有设计均是围绕这一目标来考虑的。

4.2 整体设计

我们的新系统建立在 OpenJudge 上最大的公共小组——“百练”之上，其中的原因主要由如下几点：

- 该小组中的学习者最多；
- 该小组公开，所有人均可以自由加入，体验新系统；
- 该小组题库中的所有题目均全部公开；
- 该小组题库中的题目有难有易，无论是初学者还是高手均可以找到适合自己的题目。

新系统主要包含三个子系统：多维度能力评价、个性化任务推荐和题集自动生成。用户（学习者和教师）与这些子系统之间的交互关系如图 4.1 所示。

具体说明如下：

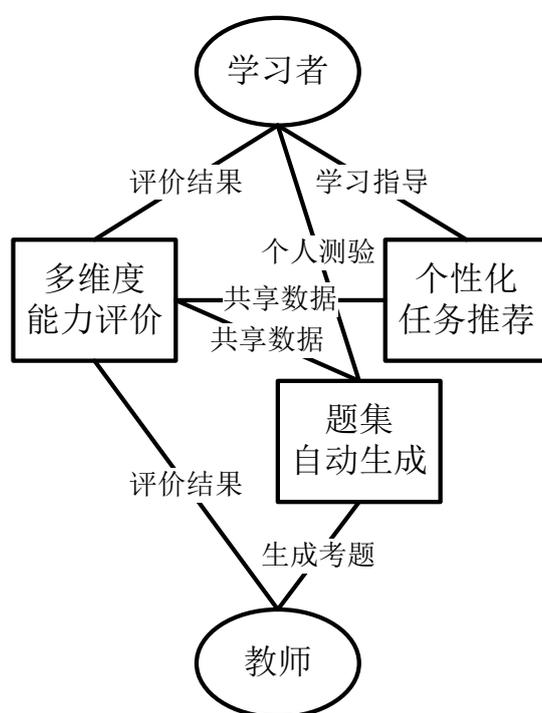


图 4.1: 新系统整体设计

- 学习者—多维度能力评价：多维度能力评价子系统通过分析学习者之前的完成的题目、比赛等结果和行为，对学习者的程序设计能力进行较为准确的评估后，反馈给学习者；
- 教师—多维度能力评价：教师可以通过多维度能力评价子系统，对自己学生的能力有更为全方位的认识，而不仅仅局限于考试中；
- 学习者—个性化任务推荐：个性化任务推荐子系统根据学习者的能力评估，在题库中选择合适的题目，以学习建议的方式推荐给学习者，并记录学习者完成任务的情况，反馈给能力评价子系统；
- 学习者—题集自动生成：题集自动生成子系统能够自动选择合适题目组成试题集，这极大方便了学习者随时随地地进行个人测试。而个人测试的结果同样也会反馈回能力评价子系统；
- 教师—题集自动生成：题集自动生成子系统极大地为教师筛选试题节省了时间，同时也大大提高了题库中题目的利用率。

等级	题数	说明
1	288	基础编程练习
2	368	简单但易错的问题
3	429	数据结构与基础算法
4	418	算法分析
5	641	程序竞赛训练
其他	123	不推荐

表 4.1: 题目分级结果

4.3 前期准备

前后花费半年多的时间，我们将百练题库中共计 2267 道题目的难度和类型全部进行了人工标注，为后续的工作打下坚实的基础。

4.3.1 题目按难度分级

在之前的改进中，我们已经将近一半的题目按难度分级完毕。现在，我们更进一步，将难度划分标准定得更为具体，并将所有题目全部分级。

我们将题目按难度从小到大依次分为 1 至 5 级，并特意设置了“不推荐”类。分级结果见表 4.1。

各级题目难度的具体划分标准如下：

- 1 级：用于让学习者逐渐熟悉编程语言以及我们的系统；
- 2 级：题目思路相对简单，但是容易犯错。可以充分训练学习者的程序设计能力和耐心，同时让他们明白良好的编程习惯的重要性——否则他们将不得不在晦涩难懂的程序中找 bug；
- 3 级：考察一些基本的数据结构和算法，这些有助于提高程序的性能；
- 4 级：考察一些高级的算法，并且需要学习者去分析时间复杂度和空间复杂

算法类型	题数	示例
基础练习	382	高精度加法
分治递归	49	汉诺塔
贪心算法	86	田忌赛马
动态规划	335	最长上升子序列
搜索算法	306	十五数码
过程模拟	237	魔兽世界
数据结构	130	伸展树
图论算法	236	最短路径
计算几何	131	凸包
数论算法	252	高斯消元

表 4.2: 题目分类结果

度，或者需要一些数学推导；

- 5 级：为希望参加程序设计竞赛的高手们准备的；
- 其他：不推荐，主要有如下几种可能：题目描述不清晰、测试数据有问题、与其他题目重复等。

4.3.2 题目按类型分类

题目分类是前期准备的另一个大项，同时也是能够进行本次改进的最重要的基础。

根据解决题目需要的算法类型，我们将题目粗略分为十类^[20]，见表 4.2。

我们特意将一些难度相近的算法类别进行合并，以避免某些类别下题目过少。另外，若一道题目有多种解法，我们优先将其按出题者的本意（或者最佳算法）进行归类。

4.4 多维度程序设计实现能力评价

4.4.1 段位制

一名程序员的理想成长道路应该是这样的：从初学开始，完成一定数量的题目后，能力提升一级，再完成一定数量的、高一个等级的题目后，能力再提升一级……直到你的能力似乎可以解决所有的问题了之后，就不能再单纯靠做题来升级，而是需要通过参加比赛——在有时间限制、有对抗的环境下，来继续提升自己的能力。当你打遍天下无敌手的时候，你就是一名优秀的程序员了，只是这还不够，一名程序员不能只满足于自己优秀，还要贡献自己的一份力量，帮助后辈们进步，同时也是锻炼自己表达能力的一种方式。

总而言之，只有通过完成题目、参加比赛以及贡献知识三个步骤，你才能成为完美的程序员，这正是我们的新系统想要达到的目的。

正因为如此，我们引入“段位制”来衡量与评价学习者目前所处的阶段——根据学习者之前的学习行为，得到一个综合的段位等级。

通过收集周围同学的学习经验、意见和建议，我们将段位划分为十级，每一级别的任务、升级条件定为表 4.3。

段位制的评价结果不仅仅反映了学习者目前所处的学习阶段，同时还能够为个性化任务推荐和个人测试建立标准。

现有的在线程序评测系统大多只采用单一的数值来反映用户的排名，这就出现了很多问题：

- 简单题和难题之间的权值差距可能过小（如果采用的是 ACM/ICPC 规则的话，甚至没有任何区别），导致大部分人为了快速提高自己的排名，选择去做简单题，而难题无人问津。这不但浪费了学习者的时间，还使得题库资源得不到充分利用；
- 对于某些高手而言，他们不屑于平时在题库中做题，却每每在比赛中摘金夺银，但该排名方法不能体现出他们的优秀；

等级	主要任务	升级条件
1	1 级题、撰写题解、比赛	完成一定数量和类型的 1 级题
2	2 级题、撰写题解、比赛	完成一定数量和类型的 2 级题
3	3 级题、撰写题解、比赛	完成一定数量和类型的 3 级题
4	4 级题、撰写题解、比赛	完成一定数量和类型的 4 级题
5	5 级题、撰写题解、比赛	完成一定数量和类型的 5 级题
6	5 级题、撰写题解、比赛	比赛积分达到一定值
7	5 级题、撰写题解、比赛、办讲座、出题	撰写题解积分达到一定值
8	5 级题、撰写题解、比赛、办讲座、出题	举办讲座积分达到一定值
9	5 级题、撰写题解、比赛、办讲座、出题	出题积分达到一定值
10	5 级题、撰写题解、比赛、办讲座、出题	N/A

表 4.3: 段位制各级主要任务及升级条件

- 在该排名系统中，学习者只需关注自己能力的提升，它不能激励用户去主动帮助他人。

在新系统中，段位仅仅是学习者的一个综合评价。为了能够全方位地评估学习者的程序设计能力，让其正确认识到自己的短板，我们还用三个完全独立的指标对其进行进一步的评价：积累分、挑战分和贡献分。

4.4.2 积累分

在积累分的计算中，我们依然是简单地将学习者解决的所有题目的权值简单相加。不过，现在的权值就不是原来那么简单了——权值的大小取决于这道题目被解决过的次数，换言之，取决于这道题目的难易程度。题目被解决的次数越少，即题目越难，权值也就越大；反之，权值就越小。

为了鼓励学习者去挑战难题，我们特意在权值-题目被解决次数的函数设计

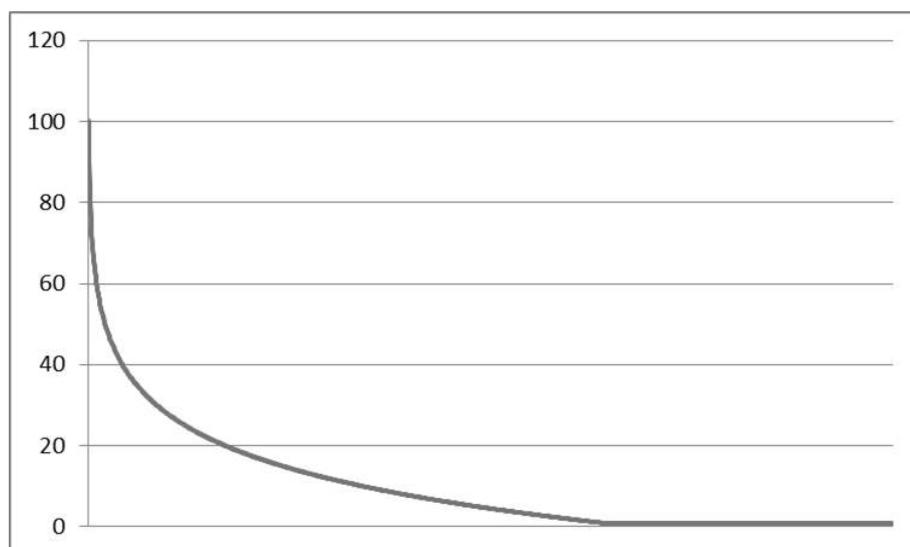


图 4.2: 权值函数图像

上有意拉开难题和简单题的权值差距。我们最终采用的权值函数为函数 4.1：

$$W(x) = \max(100 - 10 \log_2(x), 1) \quad (4.1)$$

其中 x 为该题的通过人数。

从权值函数图像（图 4.2）上来看，当学习者是唯一一名通过此题的人时，权值为100；当30名左右的学习者通过时，就只剩下50；100人通过剩下33；500人剩下10，到1000人以后，权值就只剩下1了。

4.4.3 挑战分

挑战分的目的是为了评估学习者在比赛中的表现。它同时也可以使得比赛更具竞争性，参赛者也可以从中获得更多的比赛经验。

在新系统中，学习者的挑战分将用如下方式计算：

CALCULATE-SCORE(CONTEST)

```

1  user ← {所有参加 contest 的选手（按挑战分从高到低排序）}
2  N ← length(user)
3  for i ← 1 to N
4      do score[i] ← user[i] 之前的挑战分
5  ▷ 忽略参赛者过少的比赛
6  if N < 10
7      do exit
8  for i ← 1 to N
9      do point[i] ←  $\frac{-99i+100N-1}{N-1}$ 
10 for i ← 1 to N
11     do for j ← 1 to i - 1
12         do if score[i] > score[j]
13             do point[i] ← point[i] - 1
14 sum ← 0
15 for i ← 1 to N
16     do sum ← sum + point[i]
17 for i ← 1 to N
18     do score[i] ← score[i] +  $\frac{10 \cdot N \cdot \textit{point}[i]}{\textit{sum}}$ 

```

其中,

- 第 9 行用户计算 *point* 的函数与参赛者的排名线性相关—冠军获得 100 分, 最后一名仅 1 分, 中间依次递减;
- 对于每位参赛者, 没有一位原挑战分低于他的参赛者在比赛中的成绩超过他, 他的 *point* 就会相应地减 1。第 10 行到第 13 行的部分实现了这一过程;
- 第 18 行给出了参赛者的最终得分。经过归一化处理后, 所有参赛者的得分之和将是参赛人数的十倍。需要注意的是, 如果参赛者的表现远远低于预期, 他将会获得负分。

引入这一评价机制之后, 比赛将会受到参赛用户的重视, 对抗水平也会相应

提高，比赛的价值也被更大限度地挖掘出来。

4.4.4 贡献分

除了注重自我能力的提升，用户还需要学会与他人分享自己的知识与经验。做出杰出贡献的用户，除了能够获得宝贵的贡献分外，还有机会获得由 POJ 团队直接授予的纪念勋章作为奖励。

在新系统中，我们提供了三种方式让用户贡献自己的聪明才智：

1. 撰写题解

以往学习者在做题遇到困难，无人求助之时，就会花费大量的力气在网络上搜索别人写过的解题报告，有时煞费苦心找到了一篇之后，要么看不懂，要么仔细研究之后发现这篇写的竟然是错的。这样不仅效率低，针对性不强，而且可靠性也得不到保证。

于是我们想到，既然这个系统是为了学习而设计的，那么为什么不能在学习遇到困难的时候帮一把呢？虽然有讨论区可以相互交流心得，但讨论区发帖人、发帖内容都没有受限，同样有针对性不强、可靠性得不到保证的问题。

就这样，我们在系统中开辟了一个官方的题解发布渠道（图 4.3），并要求发布者用尽可能简洁的语言描述解决这个问题的关键，称之为“一句话提示”。这个渠道的好处有：

- 可以严格限制发帖人（一定是通过此题的人）、发帖内容（只能是题解）；
- 发帖内容要求尽可能短，这样既能锻炼了发帖者的表达能力，也能避免题解过于详细，没有为用户留有足够的独立思考空间；
- 只有至少提交过一次的用户才允许查看该题题解，以免某些不自觉的用户一开始就忍不住偷看，导致思考能力得不到锻炼；
- 有贡献分的激励，可以调动用户写题解的积极性；
- 用户可以对题解评分，而这些将直接影响到发帖者获得的贡献分；为了避免



图 4.3: 提示页面

白费力气，发帖者就会自觉地对自己的题解负责，从而题解的质量可以得到保证。

2. 主办讲座

除了写写题解，资深用户还可以将自己的经验整理成一个专题报告，发送到管理员邮箱，经管理员审核之后，张贴在网站上。这不仅是对自己的锻炼、对他人的莫大帮助，更是千古留名的大好机会。

3. 比赛出题

最后一种贡献方式是为比赛出题。很多人会做起题来得心应手，但到了出题的时候就一筹莫展。出题也是一种能力，往往出一道好题，比做出几道难题都要困难。当一个人能够出题考别人的时候，他才算真正学到家了。而新系统，就为每一位用户提供了这样一个平台。

百练

百练是北京大学ACM训练和相关程序课程在线考试系统

修改设定
退出小组

管理员

Dzx guowei GMyhf
lencomputer frank david 李文
新 林舒 胡天翔

公告

03-09 踩气球活动即将开始, 敬请关注
11-03 新版百练测试赛, 丰富奖品等你来拿
11-03 test announcement1
11-01 踩气球活动即将开始, 欢迎大家参与
10-31 题目状态页面显示统计数据

成员(41465) [查看全部](#)

©2002-2013 POJ 沪ICP备12005590号-3

百练

»练习(2267题) 最新题目

题目ID	标题	通过率	通过人数	尝试人数	添加时间
4054	Cubic Eight-Puzzle	29%	9	31	2012-10-11
4052	Necklace	0%	0	7	2012-07-27
4053	Hrinity	14%	1	7	2012-07-27
4050	Chinese Repeating Crossbow	14%	1	7	2012-07-27
4051	Chess	0%	0	5	2012-07-27

»计划中的比赛

无

»已结束的比赛

2013研究生上机测试 2013-03-17 09:00:00
cs10112 Final Exam 2013-01-08 13:00:00
计概实验班上机考试 2013-01-04 13:40:00
计概上机考试练习 2012-12-28 13:45:00
cs10112 Mock Exam 2012-12-25 13:00:00

我的荣誉

勋章 **暂无勋章**

等级 **1**

积分 **170**

挑战 **0**

贡献 **0**

[创建个人测试](#)

帮助: [关于个人测试](#)

推荐任务

推荐尝试下列题目

题目ID	标题
1007	DNA Sorting
	There's
1473	Treasure Everywhere!
	Machined
1493	Surfaces
	DNA
1542	Translation
	1547 Clay Bully

推荐贡献解题思路

题目ID	标题
	Index
1109	Generation
	4007 计算字符串距离
	1014 Dividing

帮助 关于

图 4.4: 百练主页

4.5 个性化任务推荐

学习者登陆百练平台后, 右边栏会显示出“我的荣誉”和“推荐任务”(见图 4.4)。其中,“我的荣誉”即之前提到的学习者的段位等级、勋章以及三个维度的能力评价;“推荐任务”中则列出了与当前学习者等级相匹配的任务。

推荐任务会在每次打开主页时刷新并重新计算一次。在推荐题目之前, 我们会首先查看该学习者已解决的所有问题。当且仅当他已经成功解决的题数满足了这一段位上对于每种类型的题数要求, 他就会上升一级。确定学习者等级之后, 我们就可以为他推荐合适的题目, 推荐算法如下:

TASK-RECOMMENDATION(USER)

```

1  solved ← {user 已解决的题目}
2  grade ← user 之前的段位
3  recommended ← {之前推荐给 user 的题目}
4  while solved 满足 grade 的升级条件
5      do grade ← grade + 1
6          recommended ←  $\phi$ 
7  recommended ← recommended - solved
8  while length(recommended) < 5
9      do problem ← select(grade, recommended)
10     Insert problem into recommended
    
```

第 9 行的函数 “*select*(*grade*, *recommended*)” 的作用是，从题库中随机选取一个满足下列要求的题目推荐给学习者：

- 未被该学习者解决；
- 难度符合学习者的段位；
- 题目类型不予推荐列表中已存在的任何题目相同。

当然，学习者既可以从推荐的五题中任选一题来完成，也可以选择列表之外的题目。

“推荐贡献解题思路”列表将会列举出当前用户已经解决、但没有提供题解的题目，优先显示通过人数更少的题目。这个列表的目的除了方便用户定位自己没有写解题报告的题目外，还起到了一种时刻提醒用户要完成解题报告的功能，这无形中可以增加用户撰写解题报告的主动性。

4.6 题集自动生成

从用户的角度来看，比赛可以激发参与者的做题激情，这也正是 TopCoder、Codeforces 等系统如此火热的原因。但比赛最大的问题在于，它需要新的题目，

而这又恰恰是可遇而不可求的。另一方面，比赛不像考试，题目的难易、范围等不太容易预测，有时并不适合学习者。而目前大多数比赛还是侧重于竞赛，初学者无法体验比赛的乐趣。

从题库的角度来看，学习者提交的题目基本局限于很小的一部分，大部分的题目很少甚至没有人光顾。究其原因，很大程度是因为百练的主要用于学校的一些程序设计课程，而这些课程涉及到无论是作业还是考试题的数量均非常有限。这些有限的题目就成了“香饽饽”，之后的学弟学妹们也只会挑之前通过人数多的题目做。这样一来，题库的题目利用率就会非常低。

综合两方面的原因，得到一个结论：实现一个题集自动生成系统是十分必要的。

前面曾经提到过，之前的改进中已经实现了一个版本的题集自动生成子系统，但是这个子系统在实际使用中效果不甚理想，其中最大的原因就是由于没有题目的类型信息，生成的题集很容易包含多道同类型的题目，这一问题在测试乃至考试中都是不可容忍的。

现在这一问题已经成功解决了。于是，我们重新设计了新版的题集自动生成子系统，可以根据限定的题目难度分布、题目数量等自动生成一套合适的题集，并且将题目类型考虑在内。具体的题集生成算法类似题目推荐系统，这里就不再赘述。

4.6.1 个人测试

题集自动生成子系统问世之后，我们第一时间就实现了个人测试系统并上线测试。该系统旨在让学习者练习在压力与规定时间限制下完成程序设计和实现的能力，给学习者带来了另一种做题体验。同时，该系统也为学习者提供了一个自我测试和模拟考试的平台。

学习者只需要单击“创建个人测试”按钮，系统就会根据之前的做题记录，调用题集自动生成子系统生成一套合适的个人测试题集。一个个人测试时长两个



图 4.5: 关于个人测试

小时, 包含五道类型不同且该学习者未解决的题目, 其中三道与该学习者的等级相同, 另外两道“附加题”则要稍难, “附加题”的目的是为了检测学习者是否做好了升级的“准备”。个人测试提供了一种快速升级的途径—计算升级时, 学习者在个人测试中解决的题目将按双倍计算。

个人测试的具体说明可参见 <http://openjudge.cn/aboutexam.html> (图 4.5)。值得一提的是, 目前世界上各在线程序评测系统均没有实现类似功能。

4.6.2 自动安排试题

题集自动生成子系统的另一大作用就是, 辅助教师进行试题选择。在创建一个新的比赛之后, 教师可以填入对试题难度分布的限制, 然后系统便会自动填充合适的试题。当然, 如果觉得不满意, 可以任意地添加、删除或替换其中的试题。

第五章 用户行为跟踪

5.1 用户量变化分析

新系统上线以来，约有 1511 名新注册用户，而去年同期的新用户仅为 1269，增长率高达 19%。同时，近两个月的用户增长量也是这两年来最高的。

从这些新加入用户的行为来看，有将近 80% 的用户查看过与新功能有关的页面，其中又有约 40% 的用户直接使用过新功能。这说明新功能确实能够吸引新用户的加盟。

5.2 题库利用率分析

在百练题库的 2267 道题目中，多达 1149 道题目的提交量仅为 10 次或更低，这竟然占总题库的 51%，可见题库利用率之低。

而新系统上线后，单从用户完成推荐的任务情况来看，这 1149 道鲜有人光顾的题目中，就有 620 道（54%）题目有提交记录，并有 178 道（15%）题目有通过记录。从这些数据来看，题库利用率将大大提高。

5.3 用户学习效率分析

新系统上线后，活跃的用户（至少登陆过一次）有 4471 名，尝试完成任务至少一次的用户有 865 名（19%），完成至少一次任务的用户有 473 名（10%）。

在完成过任务的 473 名用户中，有 385 名（81%）用户的日均完成题目数量超过他们之前的表现，而那些表现反而变差的用户，有一部分是之前一直在解决简单的题目，导致单从数量上看数据虚高。因此，有理由相信，新系统对用户的学习效率有一定的促进作用。

5.4 题集自动生成系统使用频率分析

题集自动生成系统的实现较为复杂，是所有功能中最晚上线的，但却是所有功能里最火的。上线第一天，就有 9 名用户使用了个人测试系统，第一周的使用频率就突破 50 次，远超预期。而其中有许多用户甚至平均每 3 天就使用一次，这足以体现该系统的用户粘性。

5.5 系统安全性和稳定性分析

新系统在上线之前，已对所有页面的访问权限、各输入框的输入限制等进行了系统性的检查，并修正了部分原有版本的一些漏洞。完整上线一个多月以来，系统始终运行正常，未发现任何大问题。

通过分析数据库中的用户登陆行为和用户提交解答的记录，近一个月以来的用户同时在线最高峰值为 551，出现在 5 月 12 日 14 时 42 分左右（恰好是今年北京大学校内赛即将结束的时候），当时系统完全没有出现任何问题，可见新系统已经具备足够的承载能力，能够稳定地为广大用户提供服务。

第六章 总结与展望

前后花费了将近一年半时间的程序设计实现能力评价系统，在二次改进后正式上线。它所提供的多维度程序设计实现能力评价、个性化任务推荐及题集自动生成等主要功能，为广大程序设计学习者创造了优秀的自主学习环境，也为教师们减轻了一定的负担。

该系统吸收了众多同类在线程序评测系统的优点，并将其有机的结合在一起。同时，多维度能力评价、个人测试等功能更是目前其他系统所不具备的。

此外，我们通过对用户行为记录等内容研究，客观地评估了新系统的用户量、题库利用率、学习效率、使用频率、安全性和稳定性等多方面的指标变化，均得到了比较满意的评估结果，极大增强了我们对新系统的推广及下一步改进的信心和决心。

但是，不可否认，改进后的系统依然存在较大的上升空间。

首先是题目难度的自动标注功能。一道题目的难与否，对于不同的能力、不同经历的用户来说，是存在很大分歧的，因此需要采用一种相对客观的评价手段；对于新加入的题目，可以在很短的时间里就能收集到足够的数据来计算难度，从而大大减轻了管理者的负担；对于旧题而言，随着时代的变化，题目的难度可能也会发生改变，有了自动标注功能，就能够自动地对其进行调整。

再次，目前讲座、出题等都需要经过管理员审核，不但加重了管理负担，还是对管理员的一种考验—如果出现问题，管理员至少要承担一部分责任。我们的改进思路是采用用户评分机制，毕竟群众的眼睛是雪亮的。讲座主讲人、出题人

的贡献将直接由用户的反馈决定，因此他们必须认真对待自己的成果，才能避免自己吃力不讨好。但在这种机制下，有可能会出现问题一些不负责任的用户，甚至是一些恶意破坏的用户。初步的解决思路是借鉴 Page Rank，让每一个用户的反馈也作用在自己的积分上，这样可以避免上述情况的发生。这项工作还需要进一步的调研和建模。

总而言之，在线程序评测系统的改进工作依然任重而道远。

致谢

衷心感谢李文新老师，从一开始选题，到中期检查，再到完成整篇论文，您提出的一些思路、建议以及修改意见，均让我受益匪浅。

感谢张勤健师兄和杜仲轩师兄，如果没有你们在前期带领我上手，以及在后期给予技术上的大力支持，我很难在规定时间内完成这个项目。

感谢所有为北京大学在线程序评测系统做出贡献的前辈们，是你们让我有了发挥的舞台。

感谢实验室的伙伴们，通过在实验室的例行组会上与你们的交流与讨论，我得到了许多启发。

感谢周围的同学们，从你们身上收集到的学习程序设计的经验与教训、对原有系统的意见和建议、以及对改进后系统的反馈等，明确了我对系统的改进方向和目标。

感谢北京大学在线程序评测系统的所有用户，是你们的支持为系统提供了大量的原始数据，同时也是整个系统发展的不竭动力。

最后还要感谢我的父母以及所有关心我的人，你们的支持让我能够全身心地投入学习和研究工作中。

参考文献

- [1] 杜仲轩. 一种基于模版匹配的算法自动识别方法 [D]. 北京: 北京大学, 2012: 1, 15-17.
- [2] S.S.Skienna, M.A.Revilla. Programming Challenges [M]. New York: Springer, 2003: 1-8.
- [3] ACM/ICPC. World Finals Rules [EB/OL]. <http://icpc.baylor.edu/worldfinals/rules>. Mar 05, 2013.
- [4] Topcoder. Algorithm Competition Rating System [EB/OL]. <http://apps.topcoder.com/wiki/display/tc/Algorithm+Competition+Rating+System>. Dec 14, 2012.
- [5] Zhejiang University ACM/ICPC Team. ZOJ :: Home [EB/OL]. <http://acm.zju.edu.cn/onlinejudge>. 2013.
- [6] HDOJ ACM Team. Welcome to Hangzhou Dianzi University Online Judge [EB/OL]. <http://acm.hdu.edu.cn>. 2013.
- [7] HIT ACM/ICPC Group. HIT ACM/ICPC Group [EB/OL]. <http://acm.hit.edu.cn>. 2013.
- [8] Vijos. 关于 - Vijos [EB/OL]. <https://vijos.org/about>. 2013.
- [9] USACO. USA Computing Olympiad [EB/OL]. <http://www.usaco.org>. 2013.
- [10] ACM-ICPC Live Archive. ACM-ICPC Live Archive - HOME [EB/OL]. <http://icpcarchive.ecs.baylor.edu/>. 2013.

- [11] TopCoder. Programming Contests, Software Development, and Employment Services at TopCoder [EB/OL]. <http://community.topcoder.com/tc>. 2013.
- [12] Codeforces. Codeforces [EB/OL]. <http://codeforces.com/>. 2013.
- [13] Online Contester Team. Saratov State University :: Online Contester [EB/OL]. <http://acm.sgu.ru/>. 2013.
- [14] Timus Online Judge Team. Timus Online Judge [EB/OL]. <http://acm.timus.ru/>. 2013.
- [15] POJ. Welcome To PKU JudgeOnline [EB/OL]. <http://poj.org/>. 2013.
- [16] 杜仲轩. 一种多个在线评测系统共享评测节点的方法及其具体实现 [D]. 北京: 北京大学, 2009: 7.
- [17] 李文新, 郭炜. 北京大学程序在线评测系统及其应用 [J]. 吉林大学学报: 信息科学版, 2005(S2)
- [18] 张勤健. 一种在线评测系统中题目的自动化分类方法 [D]. 北京: 北京大学, 2011: 7-8.
- [19] OpenJudge. About [EB/OL]. <http://openjudge.cn/about.html>. 2013.
- [20] T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein. Introduction to Algorithms [M]. 2009.

附录 A 作者相关科研成果

本科生科研训练：《程序设计和实现能力的提升和评价系统研究》，2012 年 10 月

论文：A Programmer Self-training System with Programming Skill Evaluation and Personalized Task Recommendation，已通过 IEEE'13 会议的最终审稿，即将出版