UNIVERSIDAD DE VALLADOLID

Miguel A. Revilla Ramos

Dpto. de Matemática Aplicada

Thanks

- IOI I.C.
- President Zide Du
- Valentina Dagiene
- All the UVa OJ collaborators
  - Along the time
  - Around the world
- All the problemsetters, forever

Competitive Learning in Informatics: The UVa Online Judge Experience

- Miguel A. REVILLA
  - *Applied Mathematics Department*
  - *University of Valladolid, Spain*
- Shahriar MANZOOR
  - *Computer Science & Engineering Department*
  - *Southeast University, Dhaka, Bangladesh*
- Rujia LIU
  - *Department of Computer Science and Technology*
  - *Tsinghua University, Beijing, China*

Introduction
About the programming contests

- Are they an useful method for learning informatics?
  - They are probably the fastest expanding *co-curricular* activity related to computer science. They must be multiple reasons.
  - Programming lovers, whether they are secondary, high-school, or university students have a lot of choices. They are excellent as *voluntary* work.
  - It happened that most high-schools teachers of computer science courses become now more oriented to be users of applications rather than programming. **Then IOI plays a very important role here.**
  - Not only programming but algorithms, mathematics, etc.
- In my experience the answer is YES.

Introduction
Competitive vs collaborative

- Are they a positive method for learning?
  - It's clear that a programming contest is, by its own definition, a competitive activity.
  - There have been many criticisms of any kind of competitive learning and in favour of cooperative/collaborative activities.
  - In fact, many of the programming contests are team competitions and they involve a lot of collaborative work.
  - The training process involves several interesting learning strategies that have nothing to do with the real competition.
- In my opinion it depends of the training process and, of course, of the final results. I think that as far as everybody learns more they are positive.

Introduction
Are the Online Judges useful?

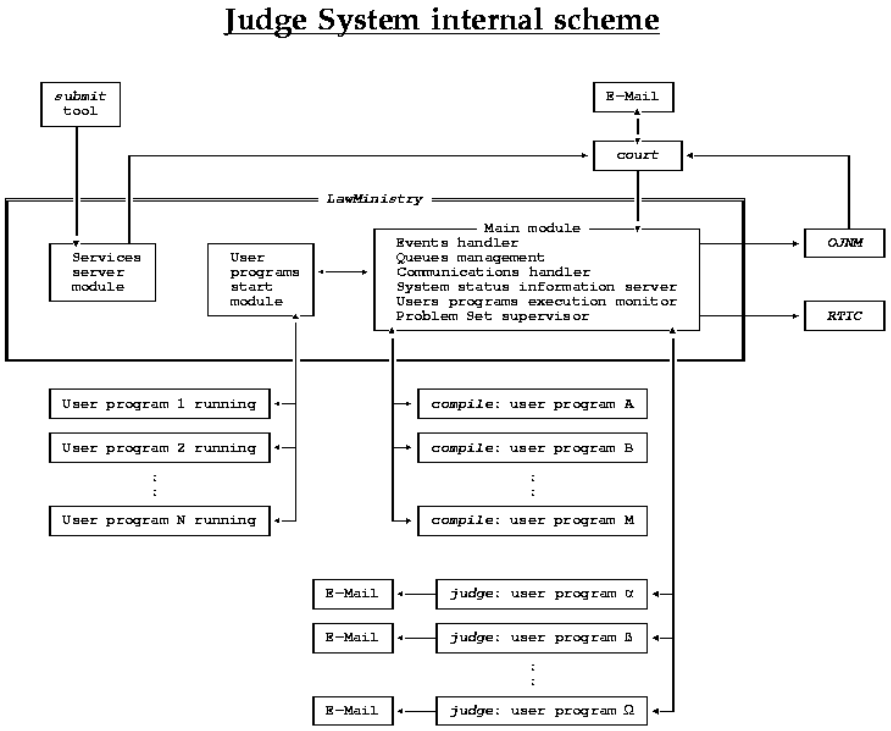
- They can provide a method for attracting interest in computer science, as they are accessible to beginning students.
  - Users from anywhere in the world can register themselves with an online judge for free and solve as many problems as they likes.
  - They can send as many solutions as they want till receiving satisfactory and/or useful information (verdict, efficiency, correction, etc.)
- They can be used for e-training, e-learning and even e-teaching.
  - They allow competitive collaboration, collaborative competition, and the probably main distinctive trait: self-testing, including self-competition via virtual contests, etc.

Agenda

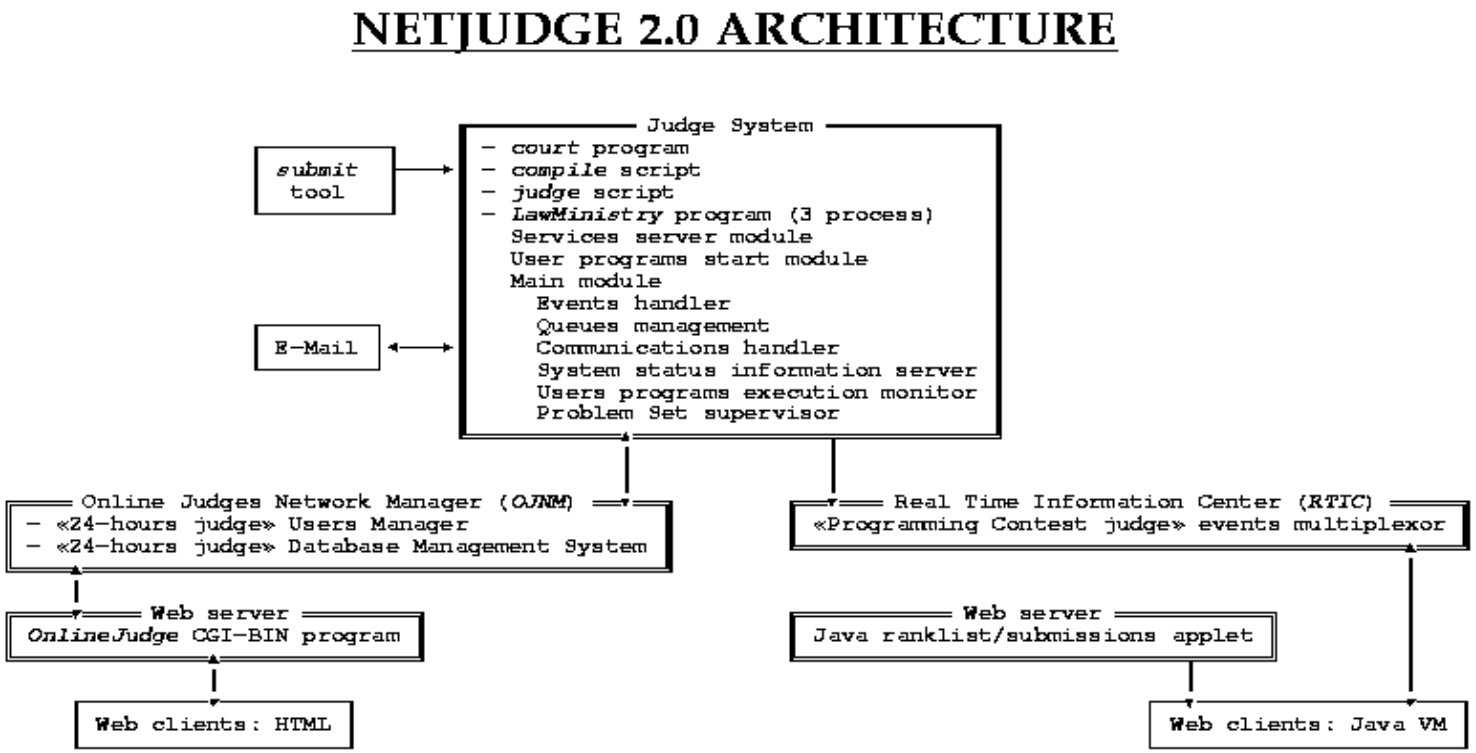
- The UVa Online Judge
- The Analysis of Statistics
- Competitive learning (practice)
- The EduJudge European Project
- IOI vs. ICPC
- Categorization of Tasks
- Conclusions

The UVa Online Judge (outside & inside faces)

- Ciriaco García



Modular structure (using Unix standard sh scripts)

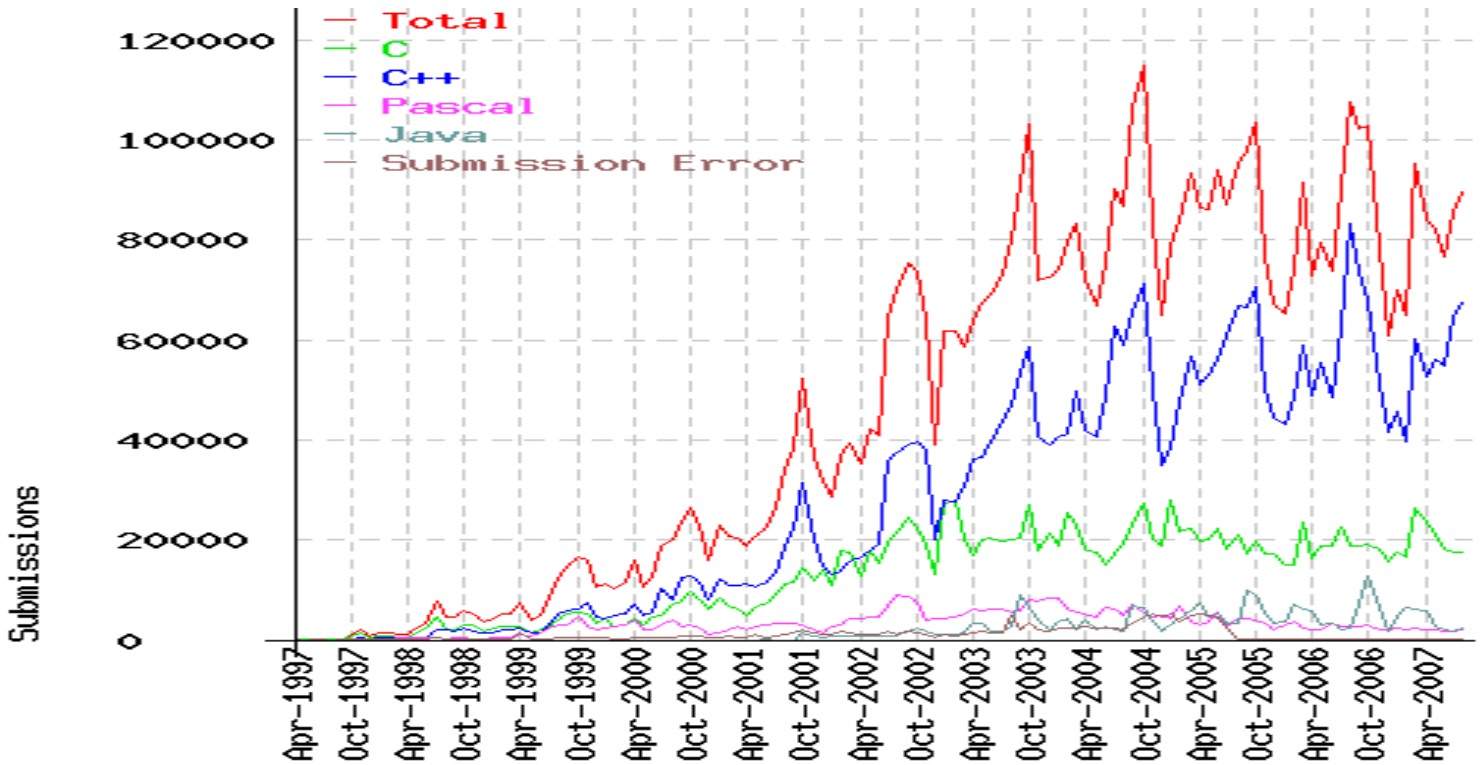


The history 'database'

Event-ID	YYYYMMDDhhmmssmmm	ST	USER	PROBL	SOURCE	MAXMEM	CPU	SG	ALGORITH
00000000	19970415143148000	AC	01000	00100	C	392	2.453	0	None
00000001	19970415160440000	SE	01001	?	C	0	0.000	0	None
00000002	19970415160531000	RE	01001	00108	C	0	0.020	11	None
00000003	19970415161102000	RE	01001	00108	C	0	0.010	11	None
00000004	19970416160933000	WA	01001	00100	C	388	2.367	0	None
00000005	19970416161547000	WA	01001	00100	C	388	5.646	0	None
00000006	19970418090211000	WA	01001	00100	C	384	5.630	0	None
00000007	19970418090254000	WA	01001	00100	C	392	5.623	0	None
00000008	19970430031739000	AC	01000	00100	C	388	2.406	0	None
00000009	19970506102716000	CE	01002	00138	C	0	0.000	0	None
05899114	20070906171446496	PE	37046	00488	C++	388	2.047	0	None
05899115	20070906171514029	WA	50251	10035	C++	0	0.047	0	None
05899116	20070906171535428	AC	48210	00841	C++	1096	0.264	0	None
05899117	20070906171600453	TL	54908	00100	JAVA	2548	10.025	9	None
05899118	20070906171617164	TL	52686	11124	C	4428	10.018	9	None
05899119	20070906171707015	WA	26416	00142	C++	404	0.121	0	None
05899120	20070906171708579	AC	47505	11049	JAVA	0	0.027	0	None
05899121	20070906171717386	RE	44804	11192	C++	0	0.002	11	None
05899122	20070906171900120	AC	48210	00841	C++	1096	0.270	0	None
05899123	20070906171943078	PE	37046	00488	C++	392	2.041	0	None

Statistics as of 2007-09-06 17:19:45 (5899124 submissions, 63351 users)

- Let's visit



Statistics by Language

(last updated 2007-09-06 17:19:45 UTC)

	Total	C		C++		Pascal		Java		Submission Error	
1997	4031	2882	71.5%	639	15.9%	2	0.0%	0	0.0%	508	12.6%
1998	42375	22440	53.0%	13859	32.7%	3819	9.0%	0	0.0%	2257	5.3%
1999	109202	39897	36.5%	42226	38.7%	22116	20.3%	0	0.0%	4963	4.5%
2000	199523	65800	33.0%	95173	47.7%	32455	16.3%	0	0.0%	6095	3.1%
2001	345305	112855	32.7%	187852	54.4%	30201	8.7%	3029	0.9%	11368	3.3%
2002	610151	212116	34.8%	305969	50.1%	62836	10.3%	13567	2.2%	15663	2.6%
2003	874762	259024	29.6%	481170	55.0%	73657	8.4%	37726	4.3%	23185	2.7%
2004	998194	243301	24.4%	607279	60.8%	69707	7.0%	41387	4.1%	36520	3.7%
2005	1050528	244620	23.3%	662432	63.1%	47978	4.6%	64898	6.2%	30600	2.9%
2006	999155	220710	22.1%	684062	68.5%	31815	3.2%	61752	6.2%	816	0.1%
2007	665898	160799	24.1%	454636	68.3%	16867	2.5%	33097	5.0%	499	0.1%
TOTAL	5899124	1584444	26.9%	3535297	59.9%	391453	6.6%	255456	4.3%	132474	2.2%

Statistics by verdict/language

(last updated 2008-07-21 02:00:27)

	TOTAL		C		C++		Pascal		Java	
TOTAL	5899124		1584445		3535293		391453		255456	
AC	1785526	30.3%	501227	31.6%	1113507	31.5%	112190	28.7%	58602	22.9%
PE	280144	4.7%	82393	5.2%	174348	4.9%	16358	4.2%	7045	2.8%
WA	2057839	34.9%	549290	34.7%	1248283	35.3%	171719	43.9%	88547	34.7%
CE	570729	9.7%	155135	9.8%	317359	9.0%	33235	8.5%	65000	25.4%
RE	429270	7.3%	138186	8.7%	281361	8.0%	8521	2.2%	1202	0.5%
TL	479287	8.1%	118812	7.5%	301382	8.5%	35765	9.1%	23328	9.1%
ML	53842	0.9%	8753	0.6%	34480	1.0%	3667	0.9%	6942	2.7%
OL	57869	1.0%	20598	1.3%	35229	1.0%	1372	0.4%	670	0.3%
RF	52141	0.9%	10051	0.6%	29344	0.8%	8626	2.2%	4120	1.6%
SE	131139	2.2%	--	--	--	--	--	--	--	--
OTHER	1338	0.0%	0	0.0%	0	0.0%	0	0.0%	0	0.0%

The most popular response sequences (based on the first 4 million submissions)

Monograph	AC	WA	CE	TL	PE	RE
Frequency	465516	324187	104952	76806	73526	62764
Di-graph	WA|WA	WA|AC	AC|AC	CE|CE	TL|TL	RE|RE
Frequency	164521	71018	49743	39732	30830	27242
Tri-graph	WA|WA|WA	WA|WA|AC	CE|CE|CE	TL|TL|TL	AC|AC|AC	RE|RE|RE|
Frequency	92545	32765	20049	14436	14203	14158
Tetra-graph	WA|WA|WA|WA	WA|WA|WA|AC	CE|CE|CE|CE	RE|RE|RE|RE	TL|TL|TL|TL	AC|AC|AC|AC
Frequency	55504	16518	11566	7947	7474	6397
Penta-graph	WA|WA|WA|WA|WA	WA|WA|WA|WA|AC	CE|CE|CE|CE|CE	RE|RE|RE|RE|RE	TL|TL|TL|TL|TL	AC|AC|AC|AC|AC|
Frequency	34775	9115	7260	4775	4172	3498
Hexa-graph	WA|WA|WA|WA|WA|WA	WA|WA|WA|WA|WA|AC	CE|CE|CE|CE|CE|CE	RE|RE|RE|RE|RE|RE	TL|TL|TL|TL|TL|TL	AC|AC|AC|AC|AC|AC|
Frequency	22475	5330	4650	2962	2433	2161

Popularity	1st	2nd	3rd	4th	5th	6th
Monograph	AC					
Frequency	465516					
Di-graph	WA|AC	CE|AC	TL|AC	PE|AC	RE|AC	
Frequency	71018	18099	10612	9213	8205	
Tri-graph	WA|WA|AC	CE|CE|AC	TL|TL|AC	CE|WA|AC	RE|RE|AC	TL|WA|AC
Frequency	32765	4685	3540	3511	2620	2423
Tetra-graph	WA|WA|WA|AC	CE|CE|CE|AC	CE|WA|WA|AC	TL|TL|TL|AC	RE|RE|RE|AC	TL|WA|WA|AC
Frequency	16518	1750	1636	1340	1158	1114
Penta-graph	WA|WA|WA|WA|AC	CE|WA|WA|WA|AC	CE|CE|CE|CE|AC	TL|TL|TL|TL|AC	TL|WA|WA|WA|AC	RE|RE|RE|RE|AC
Frequency	9115	842	827	618	573	563
Hexa-graph	WA|WA|WA|WA|WA|WA	WA|WA|WA|WA|WA|AC	CE|CE|CE|CE|CE|CE	RE|RE|RE|RE|RE|RE	TL|TL|TL|TL|TL|TL	CE|WA|WA|WA|WA|WA
Frequency	22475	5330	4650	2962	2433	2092

Main considerations

- When individual contestants make a particular type of mistakes for a problem they tend to make the same mistake again, which encourage the group to try working together for the contests. **Let's say one more time, competitive and cooperative learning in informatics are not opposite but complementary.**
- We can say that if someone gets five consecutive wrong answers then in the next submission he is four times more likely to get a wrong answer than an accepted verdict. **That means that after four or five errors, the best is to analyze carefully what happen as, probably, the mistake is not trivial.**
- It is very important the influence of the kind of mistakes in these sequences. That is, mainly, because some responses of the judge give us information about the error and others tell us nothing at all. **This is very important in order to improve our system judge to become a real learning tool, by adding new features.**

Judge response statistics for accepted problems/team only

- After 135 contests over five years
- Ratio informed vs uninformed errors is 2:1

Verdict	Percentage	Informed vs uninformed response	Informed vs uninformed errors
AC	44.16	80.89%	Not considered
PE	3.08		36.73%
WA	33.65		
TL	8.03	18.14%	18.14%
RE	3.72		
CE	6.39		
Others	0.97	Not considered	Not considered

How does practice Change Things?
Error rates based on all problems

Solve Range	AC	PE	WA	TL	RE	CE
0 – 49	23.76	4.93	36.13	8.36	8.01	12.24
50 – 99	33.81	5.57	34.18	7.33	7.54	6.35
100 – 149	35.08	6.41	33.59	6.70	7.50	5.62
150 – 199	37.02	4.95	33.01	7.07	6.90	5.70
200 – 249	37.74	5.01	32.85	7.11	6.83	5.31
250 – 299	39.90	4.60	32.41	6.89	6.16	5.17
300 – 349	40.86	4.08	32.56	7.34	5.87	4.63
350 – 399	42.03	4.30	32.21	6.51	5.97	4.49
400 – 449	41.96	4.03	32.16	6.86	6.37	4.05
450 – 499	41.82	3.65	31.50	7.10	5.98	4.68
500+	42.36	3.53	31.83	8.06	5.42	4.06

Interpretation troubles

- As people solve more problems they have less easy problems to solve (assuming that people tend to solve easy problems first).
- When someone has already solved 400 problems he has no more easy problems to solve, so his acceptance rate can go down a little. But as he is more experienced the acceptance rate does not go down but remains similar.
- **Of course, every person will always have a harder problem to solve, a new programming challenge to face in order to continuously increase his skills in informatics.**

Problems with low (less than 25%) vs. high (more than 50%) acceptance rate

Solve Range	AC	PE	WA	TL	RE	CE
0 – 49	11.09	1.71	41.73	14.62	12.43	11.52
50 – 99	17.45	2.15	42.48	13.25	12.00	6.88
100 – 149	18.98	2.69	42.13	11.85	12.44	6.16
150 – 199	20.29	2.37	41.61	12.47	10.79	6.23
200 – 249	20.86	2.46	42.17	12.78	10.15	5.77
250 – 299	23.09	2.37	41.91	12.43	9.19	5.16
300 – 349	24.24	1.94	42.17	12.46	8.92	5.01
350 – 399	24.15	2.54	42.99	11.30	9.44	4.92
400 – 449	25.61	2.33	41.32	11.42	9.51	4.47
450 – 499	27.21	2.09	38.57	12.36	8.89	5.38
500+	27.20	1.65	41.04	13.53	7.20	4.24

Solve Range	AC	PE	WA	TL	RE	CE
0 – 49	40.81	6.67	26.37	4.03	4.00	11.79
50 – 99	53.86	7.47	21.77	2.99	3.37	5.94
100 – 149	53.97	9.18	21.18	2.51	3.31	5.38
150 – 199	58.33	7.21	18.66	2.57	3.10	5.33
200 – 249	59.67	6.66	19.25	2.39	3.02	4.78
250 – 299	62.30	6.24	18.25	2.29	2.39	4.51
300 – 349	64.56	6.40	16.42	2.12	2.65	3.92
350 – 399	64.44	5.01	17.48	2.12	2.44	3.91
400 – 449	65.17	6.26	17.74	2.23	2.13	2.84
450 – 499	63.15	5.19	17.50	2.10	2.72	4.68
500+	67.73	4.31	15.46	2.22	2.33	3.97

Easy problems vs hard problems

Solve Range	AC	PE	WA	TL	RE	CE
0 – 49	27.51	4.13	35.01	7.27	7.37	12.38
50 – 99	39.32	4.37	31.79	6.19	6.51	6.57
100 – 149	40.64	6.22	31.10	5.00	6.36	5.68
150 – 199	44.10	4.11	30.16	5.51	5.59	5.68
200 – 249	45.04	4.21	29.55	5.14	5.78	5.43
250 – 299	48.07	4.09	27.60	5.07	5.18	5.22
300 – 349	49.15	3.95	27.55	5.65	4.82	4.26
350 – 399	50.11	3.75	26.57	4.95	5.29	4.63
400 – 449	50.82	4.35	26.59	4.65	5.17	4.14
450 – 499	48.69	3.87	27.68	5.26	4.71	4.42
500+	48.93	4.00	27.92	5.18	4.91	4.67
Solve Range	**AC**	**PE**	**WA**	**TL**	**RE**	**CE**
0 – 49	20.25	4.98	38.70	9.02	9.05	11.26
50 – 99	30.00	5.27	36.92	7.58	8.74	6.08
100 – 149	33.15	7.11	35.39	6.73	7.30	5.39
150 – 199	29.28	4.91	37.41	8.16	7.59	6.24
200 – 249	30.36	6.02	35.89	7.75	7.89	6.07
250 – 299	33.81	4.59	36.33	7.80	6.36	5.71
300 – 349	32.71	3.75	37.60	8.54	6.54	6.27
350 – 399	30.29	4.38	39.85	8.78	6.28	5.04
400 – 449	30.71	3.54	38.91	8.36	7.80	5.32
450 – 499	32.92	3.02	36.12	10.24	8.47	4.42
500+	33.73	2.59	39.12	10.59	4.80	4.34

Difficulty vs acceptance rate (assumed the same time up)

- Just heuristics (experimental difficulty)
- Easy: problems that have been accepted most in numbers
- Hard: problems that have been accepted least in numbers
- A difficult problem can have good acceptance rate
  - Suppose problem 1001 is submitted 1000 times and the accepted number is 200
  - Suppose problem 1002 is submitted 100 times and the accepted number is 80
  - So 1001 is an easy problem with low acceptance rate and problem 1002 is a hard problem with high acceptance rate

Comments and …

- Usually, by 'programming ability' people means coding, debugging and testing. Though, these individual abilities greatly affect cooperative works too (it's easy to suppose that many of our users work in group, being a team or not).
- Most people got started by solving easy problems. Here, by easy problems, we mean the problems in which you only need to do what you're asked to do, i.e. a direct implementation of the problem description.
- ICPC, IOI and most of the existing contests concentrate on problem solving rather than software engineering

… and tips

- When getting started, practice is much more important than theory.
- Everyone is encouraged to program as much as he can, as long as enthusiasm is perfectly kept.
- Trying to solve more problems is good, but the quantity is not the most important thing.
- It is better to solve many problems of various kinds and difficulty.
- So users will be challenged with problems that are more interesting and difficult, becoming a great contestant.

OVERVIEW OF THE EDUJUDGE PROJECT

**This project has been funded with support from the European Commission.
This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein**

FRAMEWORK

- The EduJudge project is funded with the support of the **Lifelong Learning Programme** of the **European Union**.
- It is part of the Transversal Programme – Key Activity 3: *Development of ICT-based content and services.*

PARTNERSHIP

- **CEDETEL** (Coordinator) - Spain
- **University of Valladolid** – Spain
- **University of Porto** – Portugal
- **KTH Royal Institute of Technology** – Sweden
- **Institute of Mathematics and Informatics** - Lithuania

AIMS AND OBJECTIVES

- **MAIN OBJECTIVE:** to integrate **UVA Online Judge** (online-judge.uva.es) into an **effective educational environment**, in order to satisfy the users' demand of a greater pedagogic character and, in this way, to facilitate the use of Online Judge in the official courses offered in the areas of **mathematics and programming**.
- **To give Online Judge a greater pedagogic character** designing an intelligent system and classifying problems into different levels of difficulty.
- **To create a community of teachers and students** at European level to share Knowledge and experiences.

WORK PACKAGES

MNGT 1
COORDINATION

CEDETEL

ALL

PREP 1
REPOSITORY OF PROBLEMS

U. PORTO
ALL

PREP 2
EXTENSION ONLINE JUDGE

UVA
KTH, UP

PREP 3
INTEGRATION ELEARNING PLATFORM

UVA
ALL

QPLN 1
PILOT EXPERIENCES

IMI
ALL

QPLN 2
QUALITY & EVALUATION

CEDETEL
ALL

DISS 1
DISSEMINATION

CEDETEL
ALL

EXP 1
EXPLOITATION

UVA
ALL

PREP 2 Extension, improvement and optimization of the Online Judge evaluation engine

The result of this work package is the Online Judge improved with the following functionalities:

- Solution quality evaluation (T1): the system will analyse the results produced by the solution proposed by the student, and will determine its percentage of success (**grading system**). Thanks to this, the student will be able to detect and solve eventual errors during the development of the solution.
- Generic judge engine (T2): the system will support several problem formats (**batch and reactive**), allowing different kinds of learning approaches.
- Automatic testcase generation and validation (T3): the automatic generation of testcases will allow different levels of difficulty in the solving of the problems.

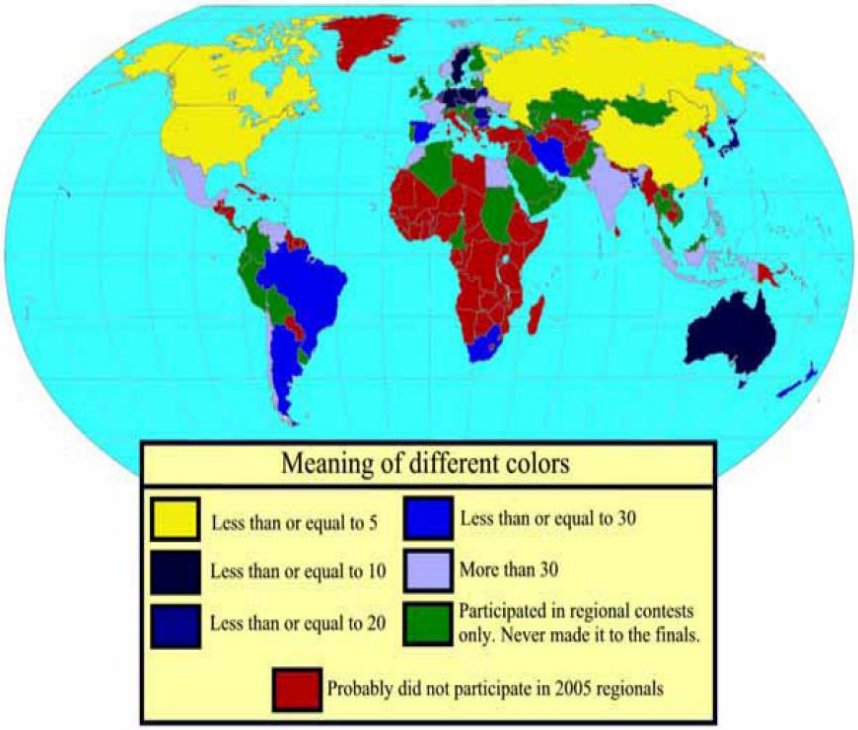
TARGET GROUPS

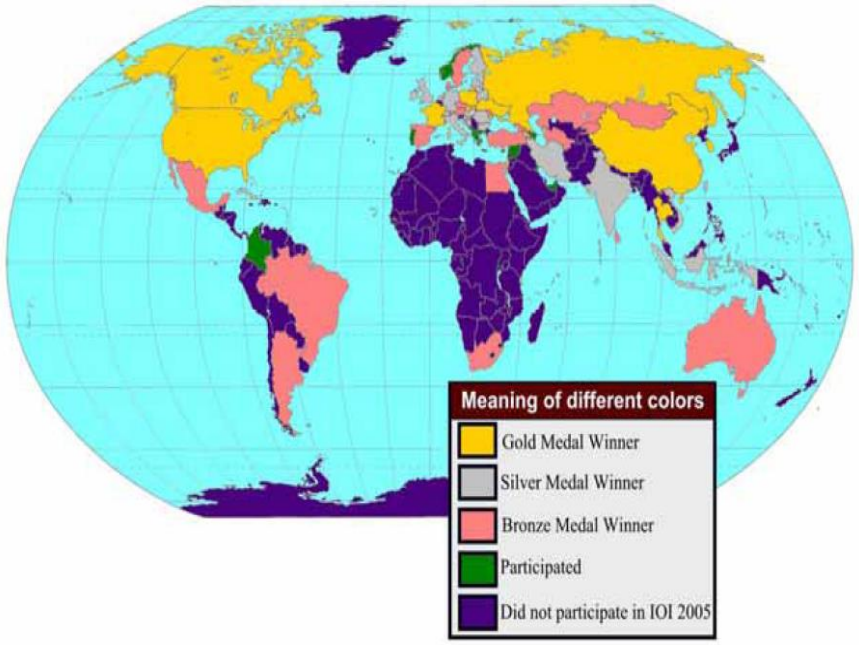
- University sector and post degree, for several technical and science degrees.
- Secondary school, for computing field.
- Teachers

IOI vs. ACM-ICPC
Is the convergence possible?

- I don't mean about the contests but about the philosophy in order to implement EduJudge allowing
  - E-training
  - E-learning
  - E-teaching
  - Several methods of grading
  - Several types of problems
  - Several styles of contest
  - Is there a 'meeting point' online contest possible?

ICPC average performance (1998-2005) vs IOI medal distribution (2005)





The first and main problem: the issue of grading

- Automated grading is almost impossible without a 'bit' of human participation.
- It's evident that this is an additional trouble for the problem setters
  - Test cases need to be more carefully selected to produce a gradual punctuation correlated with the correctness.
  - Even the description of the problems need to be analyzed to make it reasonable to claim that a program is 50% correct.
- Any kind of conventional grading system is closer to the competitive learning objective of the EduJudge project than the 0/1 approach of ICPC.

Positive sides

- IOI
  - It allows partial marking unlike the 0/1 approach of ICPC.
  - It requires the contestants to solve only three problems in five hours.
  - The speed of a contestant is not a strong factor.
- ICPC
  - It gives real time feedback to contestants and also gives some credit to the speed.
  - The three member team structure promotes the cooperative learning added to the competitive situation.

Then a hybrid ideal contest (this is not a proposal, but a mind)

- Should have partial credits like IOI and also real time feedback like ICPC.
- Possibly informs the contestant which test cases match (only the serial of test case) and which don't.
- There are a bunch of questions
  - How a contest model can give partial credit and allows infinite times submission.
  - How can we prevent poorly written solutions to get good scores?
  - How weak coders will get marks if there is no lenient rule like the classical 50% rule?
  - What about the team composition? One, two, three?
  - How can a model be fair and simple enough?

We tested these kinds of events in our Hosting Contest Service …

- Judge response statistics based on accepted problems/team only

Submission Serial	Cumulative Acceptance Percentage	Acceptance Percentage	Cumulative Number of Acceptance	Submission Serial	Cumulative Acceptance Percentage	Acceptance Percentage	Cumulative Number of Acceptance
1	53.622455	53.622455	24358	11	98.908090	0.305999	44929
2	72.686846	19.064392	33018	12	99.119428	0.211337	45025
3	82.875069	10.188222	37646	13	99.317556	0.198129	45115
4	88.920198	6.045129	40392	14	99.493671	0.176114	45195
5	92.631811	3.711613	42078	15	99.583930	0.090259	45236
6	94.996147	2.364337	43152	16	99.667584	0.083654	45274
7	96.398459	1.402312	43789	17	99.749037	0.081453	45311
8	97.367089	0.968630	44229	18	99.806274	0.057237	45337
9	98.093561	0.726472	44559	19	99.856907	0.050633	45360
10	98.602091	0.508531	44790	20	99.894331	0.037424	45377

… but we still need to check our online contest with real ones.

- Judge response statistics ignoring first four uninformed responses and allowing maximum eight informed errors

Submission Serial	Cumulative Acceptance Percentage	Acceptance Percentage	Cumulative Number of Acceptance	Submission Serial	Cumulative Acceptance Percentage	Acceptance Percentage	Cumulative Number of Acceptance
1	63.077600	63.077600	28653	10	99.225096	0.323610	45073
2	80.061640	16.984040	36368	11	99.392405	0.167309	45149
3	88.453495	8.391855	40180	12	99.509081	0.116676	45202
4	93.021464	4.567969	42255	13	99.643368	0.134287	45263
5	95.601541	2.580077	43427	14	99.720418	0.077050	45298
6	97.076500	1.474959	44097	15	99.795267	0.074849	45332
7	97.932856	0.856357	44486	16	99.843698	0.048431	45354
8	98.507430	0.574573	44747	17	99.876720	0.033021	45369
9	98.901486	0.394056	44926	18	99.898734	0.022014	45379

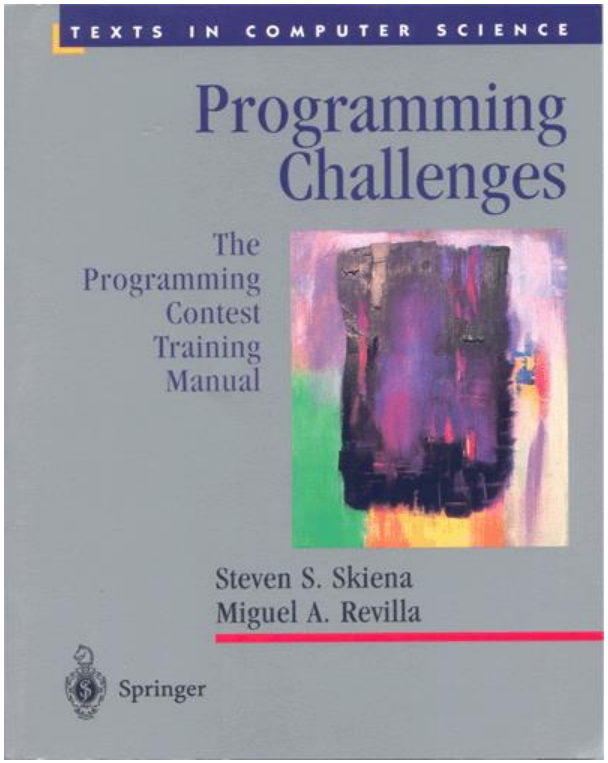
Categorization
Solving methods and/or topics

- Serious solvers are not interested in doing
- Is very useful, specially for beginners and teachers
- However, maintenance is a really hard task
- The solution could be a little list for each category
  - Classical problems
  - Very specific exemples
- Even though there is an almost standard universally accepted list, the experience shows us that the contribution of the users must be managed if we want to prevent a real chaos.
- **Let's take a look to the main wiki I know about**

Difficulty level (even worse)

- The opinion is very subjective
  - Depends of the expertise of the people making the decision and, of courser, of the user
- Probably most the people agree about easy and hard
  - Of course, about trivial and very hard
  - But what about medium levels?
- A very important detail for the learning efficiency
- Must be very clear that the difficulty level is a relative concept to prevent traumatic experiences to the users.

Skiena & Revilla, 2003
(to enjoy learning informatics for free)



- There are many distinct pleasures associated with computer programming. (…)
- The games, puzzles, and challenges of problems from international programming competitions are a great way to experience these pleasures while improving your algorithms and coding skills.

- ***Programming Challenges***.
The Programming Contest Training Manual

UVa Online Judge

Conclusions

- **Competitive learning in informatics**, as we understand it in the present paper (training to participate in programming contests by using online judges and taking part in internet contests) **can be an adequate method to learn algorithms and programming**, as it is free of the most frequent criticisms that many other methods have.
- It doesn't the matter whether the contest is individual or by teams, **most of the work to do is self-competitive** as well as cooperative. The 'learning' criteria requires that the common work and the individual effort must go together.
- Many students qualify for the big events of ICPC, IOI and TopCoder but many more students never qualify for a bigger event, but behind this tangible failure, **they become better programmers and thinkers**, which may in future help them to become something special.

Thank you for your attention
Questions?

e-mail:revilla@mac.cie.uva.es
e-mail:shahriar_manzoor@yahoo.com
e-mail:rujia.liu@gmail.com
e-mail:carlos.marce@gmail.com
e-mail:yo@miguelrevilla.com
e-mail:problemset@acm.uva.es
e-mail:contest@acm.uva.es

Thank you for your attention
Questions?

- *e-mail:revilla@mac.cie.uva.es*
- *e-mail:shahriar_manzoor@yahoo.com*
- *e-mail:rujia.liu@gmail.com*
- *e-mail:carlos.marce@gmail.com*
- *e-mail:yo@miguelrevilla.com*
- *e-mail:problemset@acm.uva.es*
- *e-mail:contest@acm.uva.es*

References

- UVa Online Judge. *Online Judge and Contest system developed by the University of Valladolid (Spain)*
  http://online-judge.uva.es/problemset
- CLI. *Competitive Learning Institute at the Baylor University of Texas (USA).* http://icpcres.ecs.baylor.edu/onlinejudge
- Pi Algorithmist. *The Algorithmist is a resource dedicated to anything algorithms.* http://www.algorithmist.com/index.php/Main_Page
- Igor's UVa tools. http://shygypsy.com/acm/
- Skiena, Steven S., M. A. Revilla (2003). *Programming Challenges. The Programming Contest Training Manual.* Springer-Verlag, New York.
- Manzoor, S. (2006). *Analyzing Programming Contest Statistics*. CLIS, San Antonio.
- Liu, R. (2008). *Training ICPC Teams: A Technical Guide*. CLIS, Banff.