Analyzing Programming Contest Statistics

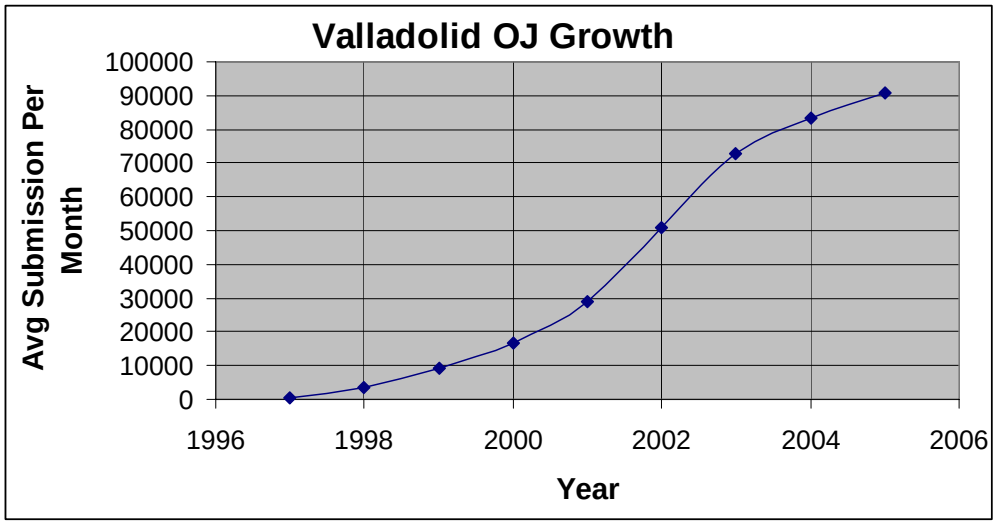
Shahriar Manzoor
(shahriar_manzoor@yahoo.com **)**
Southeast University, Dhaka, Bangladesh
Member of Elite Problemsetters' Panel
ACM Valladolid Online Judge, Online Contest Wing
(http://online-judge.uva.es/contest/)

In this paper we will try to analyze a large database of submission of contestants from different parts of the world and provide some important results which was never presented before in this manner and at this scale. These results may enable us to find out what aspects of programming contests need to be changed to make it more meaningful, how practice and experience improves the performance of a contestant and it will also create new openings on which we can continue our study in future. It will also help us to identify the geographic locations which are not lightened by programming contest and then we can take initiatives for those regions. At the end of this paper we will try to put together some suggestions to improve programming contest by making IOI and ICPC more similar.

1. Introduction:

Programming contest is probably the fastest expanding co-curricular activity related to computer science. The main reason is because arranging a programming contest does not require too much change in infra structure. Generally each type of programming contest has an annual international event, but teams for these contests are selected via many preliminary contests which takes place in national level, regional level or sometimes as online contests. While ICPC (ACM International Collegiate Programming Contest, Web: http://icpc.baylor.edu), IOI (International Olympiad for Informatics, Web: http://**olympiad**s.win.tue.nl/**ioi/** and http://ioinformatics.org/) and Topcoder (www.topcoder.com) are the three most prominent programming contests of the world, there are some other web sites which provide adequate training materials for these contests. These web sites are known as Online Judges to contestants and most of these sites are prepared based on the rules and restrictions of ICPC – the largest programming contest for University Students.

Some well known online judges are ACM Valladolid Online Judge or UVaOJ (http://acm.uva.es/p), Zhejiang University Online Judge (http://acm.zju.edu.cn/), Saratov State University Online Contester (http://acm.sgu.ru) and Ural State University Problem Set Archive (http://acm.timus.ru). Of these sites the Valladolid Online Judge is the oldest Online Judge and arguably the most popular Online Judge as well. In started in 1997 and since then it has become more and more popular keeping pace with the increasing popularity of programming contest all over the world the world. The following chart shows how the submission per month (on average) has grown in UVa judge from its beginning. This site has received about 4089635 submissions from different corners of this world from its beginning till October, 2005. We will use this huge number of submissions to analyze different aspects of programming contest. To imagine how huge this number of submissions is, let's assume there are 8 problems and 60 teams in a programming contest and all the teams solve all problems with 4 submissions per problems. So the total number of submissions is 60*8*4=1920. So we can assume (relaxed assumption) that a five hour programming contest has 1920 submissions in total. So the number submission of UVa OJ is equivalent to 4089635/1920~2130 real-time programming contests. Before analyzing the huge data, we will give you a short introduction on online judge.



2. What is UVa Online Judge?

An online judge is in general a server which contains problem description of different contests. The judge also contains the judge data to judge most of the problems. All the problems have an unique ID. Any user from anywhere in the world can registrar himself with an online judge for free and solve any problem he likes – he just has to mention the problem ID and his own membership number while submitting the problem. Most online judges allow its users to submit solutions in at least three languages C/C++, JAVA and PASCAL. Although C and C++ are two different languages but it is hard to distinguish between them because they are supported by the same compiler, they have many things in common, contestants use both the features of C and C++ when it is more convenient and many contestant codes in C and submits them as C++ program to avoid some type conversion complications of C. For example some contestants use STL and C++ but uses printf() function to produce output as output formatting is often much easier with printf() function.

UVa online judge is one such online judge whose main responsible person Prof. Miguel A. Revilla has received DeBlasi Award from ACM for its outstanding contribution in popularization and training of programming contest. It has about 1800 problems to solve and about 64000 users world wide. It is also mentioned before that the site has received around 4 million submissions up to October 2005, which are the primary data for our analysis. When a user submits his solution for a problem, the online judge tests it with an specified input and matches its output with the specified output to test its correctness. Of course some problems have more than one solution and to judge those problems some special judge programs are used. According to the outcome of this test the user is given any one of the following responses

Table 1: Meaning of different verdicts of the judge.

Short Notation	**Meaning**	**Detailed Meaning**	**Percen-tage** (24 Hour Online Judge)	**Percen-tage** (Realtime)	**Percen-tage** (24 Hour Online Judge Only with Online Contest Problems)
AC	Accepted	The output of the program matches correct output.	30.36	24.09%	32.55%

PE	Presentation Error	The output is correct but it produces some extra space or blank line.	5.18%	2.20%	3.01%
WA	Wrong Answer	The output of the program does not match the correct output.	36.15%	43.81%	37.34%
TL	Time Limit Exceeded	The program does not terminate within the specified time limit.	8.10%	14.16%	8.60%
CE	Compile Error	The program does not compile with the specified language's compiler.	9.72%	8.49%	8.74%
RE	Runtime Error	The program has crashed.	7.85%	5.30%	7.01%
ML	Memory Limit Exceeded	The program requires more memory to run than what the judge allows.	0.84%	0.92%	1.06%
OL	Output Limit Exceeded	The program produces more than 4 MB output within the time limit.	0.96%	0.38%	0.88%
RF	Restricted Function	The program uses some system function call or tries to access files.	0.84%	0.64%	0.81%
Others	Uncommon verdicts	Some rare verdicts produced by the judge which are not that important in the context of this report	0.33%	Ignored	Ignored

The last column shows which error occurs at what percentage. So the percentage of "accepted" verdict is 30.36 % means of the total verdicts produced by the judge 30.36% are accepted. Another interesting things from this table is that the principle verdicts produced by the online judge are Accepted (AC), Wrong Answer (WA), Compile Error (CE), Time Limit Exceeded (TL), Run Time Error (RE) and Presentation Error (PE). So in this article we will focus more on these six types of responses. Also these five errors are more common in realtime contests. Errors like SE, ML, OL, RF etc are generally not considered in realtime contest or considered within the primary five errors. For example in most contests Memory Limit Exceed is considered within Compile Error, Output Limit Exceeded is considered within Time Limit Exceeded or Wrong Answer.

In the next section we will consider different aspects of programming contest based on the huge submission history of UVa Online Judge. The submission history of UVa Online Judge is so big that these can actually represent the general characteristics or errors, successes in a programming contest.

3. How does practice change things?

The table below shows the error rate of people with different experience. The left most column actually describes the experience of the user that is being considered. Each of the next six columns are actually reserved to display the rates of six major judge responses in a programming contest. For example the third row of the table below says that the contestants who have solved 25 or more problems has 35.40% acceptance rate, the rate for wrong answer is 33.75 and so on.

Table 2: Based on all problems

User Type	AC	PE	WA	TL	RE	CE
0 or more	29.33	5.01	34.92	7.83	7.58	9.39
25 or more	35.40	5.28	33.75	7.31	7.22	5.88
50 or more	37.00	5.12	33.26	7.09	7.00	5.45
75 or more	37.79	5.03	33.01	7.04	6.87	5.21
100 or more	38.52	4.91	32.82	6.98	6.75	5.03
125 or more	39.29	4.59	32.65	7.04	6.56	4.88
150 or more	39.96	4.28	32.49	7.10	6.43	4.78
175 or more	40.51	4.17	32.39	7.09	6.32	4.61
200 or more	40.96	4.06	32.32	7.10	6.27	4.46
225 or more	41.51	3.88	32.25	7.10	6.18	4.31
250 or more	41.93	3.77	32.16	7.10	6.11	4.21
275 or more	42.17	3.65	32.14	7.12	6.12	4.09
300 or more	42.48	3.55	32.09	7.16	6.09	3.95
325 or more	42.76	3.46	32.01	7.15	6.11	3.83
350 or more	42.88	3.41	31.98	7.11	6.15	3.78
375 or more	43.14	3.28	31.90	7.19	6.15	3.67
400 or more	43.08	3.21	31.92	7.25	6.19	3.62
425 or more	43.14	3.14	31.95	7.30	6.18	3.56
450 or more	43.34	3.02	31.86	7.34	6.15	3.52
475 or more	43.43	2.93	31.98	7.41	6.15	3.38
500 or more	43.68	2.88	31.94	7.40	6.19	3.26
Difference	14.35	-2.13	-2.98	-0.43	-1.40	-6.12

The table below shows the same results from a different viewpoints. The second row shows the different error rates for the people solving less than 50 problems, the fourth row shows different error rates for the people solving 100 to 149 (inclusive) problems.

Table 3: 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

The above tables indicates that with practice acceptance rate increases a lot and also compile error decreases a lot but surprisingly wrong answer and TL percentage does not change that much. So does this indicate no matter how experienced you are you can always get wrong answer?

The above table can have a problem, as people solve more problems they have less easy problems to solve (assuming that people tend to solve easy problems first) so when someone has already solved 400 problems he has no more easy problems to solve and so his acceptance rate can go down a little but as he is more experienced the acceptance rate does not go down but it remains similar.

In the table below we have put the same results but this time only based on the problems with low acceptance rate.

Table 4: Based on problems with low (less than 25%) acceptance rate.

Solve Range	AC	PE	WA	TL	RE	CE
0 or more	15.81	1.98	41.85	13.56	11.70	8.79
25 or more	19.64	2.20	42.12	12.99	11.21	6.13
50 or more	21.05	2.27	41.99	12.39	10.90	5.76
75 or more	21.87	2.31	41.86	12.20	10.66	5.47
100 or more	22.48	2.31	41.79	12.04	10.46	5.31
125 or more	23.05	2.26	41.85	12.08	9.99	5.14
150 or more	23.78	2.17	41.66	12.11	9.73	5.00
175 or more	24.37	2.16	41.62	12.07	9.48	4.81
200 or more	24.82	2.11	41.67	12.01	9.41	4.63
225 or more	25.40	2.08	41.56	11.94	9.32	4.44
250 or more	25.85	2.02	41.54	11.81	9.22	4.34
275 or more	26.09	1.95	41.63	11.74	9.18	4.23
300 or more	26.48	1.94	41.46	11.66	9.23	4.15
325 or more	26.69	1.94	41.34	11.62	9.31	4.04
350 or more	26.98	1.94	41.30	11.48	9.30	3.95
375 or more	27.40	1.89	41.08	11.52	9.25	3.81
400 or more	27.55	1.82	40.96	11.52	9.27	3.76
425 or more	27.70	1.77	40.99	11.51	9.28	3.69
450 or more	27.94	1.72	40.89	11.54	9.22	3.62
475 or more	27.88	1.69	41.18	11.50	9.26	3.45
500 or more	28.08	1.65	41.32	11.38	9.28	3.28

Table 5: Based on problems with low (less than 25%) 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

In the table below we have put the same results but this time only based on the problems with high acceptance rate.

Table 6: Based on problems with High (more than 50%) acceptance rate

Solve Range	AC	PE	WA	TL	RE	CE
0 or more	48.43	6.90	23.46	3.38	3.57	8.79
25 or more	56.31	7.22	20.59	2.75	3.12	5.55
50 or more	57.83	7.18	19.86	2.58	3.04	5.10
75 or more	58.92	7.15	19.20	2.50	2.93	4.90
100 or more	59.87	7.03	18.88	2.36	2.88	4.67
125 or more	61.04	6.72	18.34	2.34	2.78	4.50
150 or more	62.52	6.06	17.84	2.30	2.68	4.35

175 or more	63.36	5.85	17.64	2.22	2.58	4.12
200 or more	64.01	5.65	17.55	2.20	2.53	4.00
225 or more	64.92	5.42	17.17	2.17	2.42	3.86
250 or more	65.49	5.30	16.96	2.13	2.37	3.73
275 or more	65.94	5.19	16.83	2.10	2.37	3.53
300 or more	66.49	5.01	16.56	2.08	2.36	3.49
325 or more	66.88	4.81	16.50	2.08	2.29	3.41
350 or more	67.01	4.63	16.60	2.07	2.28	3.37
375 or more	67.50	4.56	16.52	2.03	2.24	3.22
400 or more	67.71	4.53	16.36	2.06	2.23	3.22
425 or more	67.84	4.36	16.19	2.01	2.29	3.32
450 or more	68.36	4.08	16.00	2.01	2.26	3.32
475 or more	69.16	3.89	15.66	2.08	2.15	3.14
500 or more	69.67	3.80	15.62	1.99	2.14	2.98

Table 7: Based on problems with High (More than 50%) acceptance rate

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

By generating such tables for different types of problems then we can design a problemset that will give experienced coders less or more advantages (the one that this required). Also all these can make programming contest more interesting in future. If we want to give rating to contestants, find out the probability of their winning based on the problemset and previous history these types of statistics can help. Also coaches can can find from these statistics whether his team is performing better than average or less than average. The tables for easy and hard problems are also given below:

Table 8: Based on easy problems

Solve Range	AC	PE	WA	TL	RE	CE
0 or more	33.13	4.32	33.30	6.63	6.87	9.92
25 or more	40.57	4.65	31.31	5.84	6.32	6.24
50 or more	42.59	4.64	30.40	5.57	6.03	5.77
75 or more	43.53	4.77	30.01	5.39	5.86	5.46
100 or more	44.63	4.81	29.54	5.18	5.74	5.27
125 or more	45.81	4.39	29.14	5.26	5.51	5.12
150 or more	46.87	4.02	28.66	5.28	5.39	5.04
175 or more	47.48	3.97	28.38	5.22	5.30	4.91
200 or more	48.18	3.98	27.95	5.17	5.29	4.73
225 or more	48.83	3.96	27.72	5.07	5.22	4.56
250 or more	49.49	3.89	27.28	5.18	5.08	4.44
275 or more	49.69	3.88	27.36	5.14	5.08	4.27
300 or more	50.02	3.81	27.16	5.22	5.05	4.15
325 or more	50.16	3.76	27.00	5.30	5.08	4.09

350 or more	50.31	3.77	27.03	5.08	5.12	4.12
375 or more	50.25	3.82	27.14	5.13	5.14	4.01
400 or more	50.37	3.77	27.18	5.12	5.06	3.95
425 or more	49.99	3.75	27.48	5.20	5.05	3.94
450 or more	50.23	3.59	27.37	5.27	5.03	3.89
475 or more	50.32	3.45	27.44	5.36	5.12	3.84
500 or more	50.74	3.50	27.27	5.28	5.13	3.71

Table 9: Based on easy 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

Table 10: Based on hard problems

Solve Range	AC	PE	WA	TL	RE	CE
0 or more	25.14	5.04	37.92	8.56	8.51	8.71
25 or more	30.09	5.03	37.22	8.11	8.28	5.78
50 or more	31.28	5.12	36.94	7.97	7.84	5.50
75 or more	31.50	5.13	36.83	8.19	7.63	5.34
100 or more	31.86	5.06	36.95	8.15	7.43	5.23
125 or more	30.93	5.06	37.43	8.50	7.46	5.22
150 or more	31.31	4.18	37.62	8.76	7.48	5.16
175 or more	31.67	4.09	37.64	8.86	7.39	4.95
200 or more	31.92	3.96	37.68	8.94	7.45	4.84
225 or more	32.06	3.61	37.94	9.25	7.41	4.62
250 or more	32.36	3.39	38.18	9.27	7.33	4.50
275 or more	32.14	3.29	38.23	9.46	7.52	4.41
300 or more	32.07	3.16	38.54	9.55	7.52	4.26
325 or more	32.17	3.07	38.53	9.74	7.59	3.94
350 or more	31.95	3.04	38.72	9.74	7.71	3.87
375 or more	32.08	2.86	38.67	9.92	7.89	3.71
400 or more	32.25	2.80	38.51	9.92	7.96	3.66
425 or more	32.45	2.73	38.58	10.00	7.91	3.53
450 or more	32.53	2.67	38.44	10.20	7.99	3.37
475 or more	32.57	2.61	38.52	10.33	7.90	3.26
500 or more	32.47	2.61	38.82	10.19	7.91	3.20

Table 11: Based on hard problems

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

4. Drop out rate!!!

Programming contest is not something that is very easy and generally people with average intelligence struggle in this field. It is quiet impossible to find out how the drop out rate of programming contest (How many people hopes to participate in programming contest but then never does so). Of the 65000 users so far in Valladolid Site we have found 739 people who have submitted 20 or more times but have failed to get one problem accepted. People who have got something accepted probably would have learnt something, but these people have left the arena without probably learning anything or probably learning something bitter (They are not fit for contest). There is one person who have 2945 times and has not got anything accepted and another person who has submitted 540 times but have not got anything accepted. But these are very extreme cases and there are not too many like this.

5. Where is programming contest more popular?

With the nine years submission history of UVa we have found which regions have used this site more, and we can safely assume that the people of regions which uses this site more like programming contest but the vice versa may not always true. We have defined the acceptance ratio (ACR) for a country C as:

$$\text{ACR}=\frac{Number\ of\ \ submissions\ \ from\ country\ C}{Population\ of\ \ Country\ C\ in\ million}$$

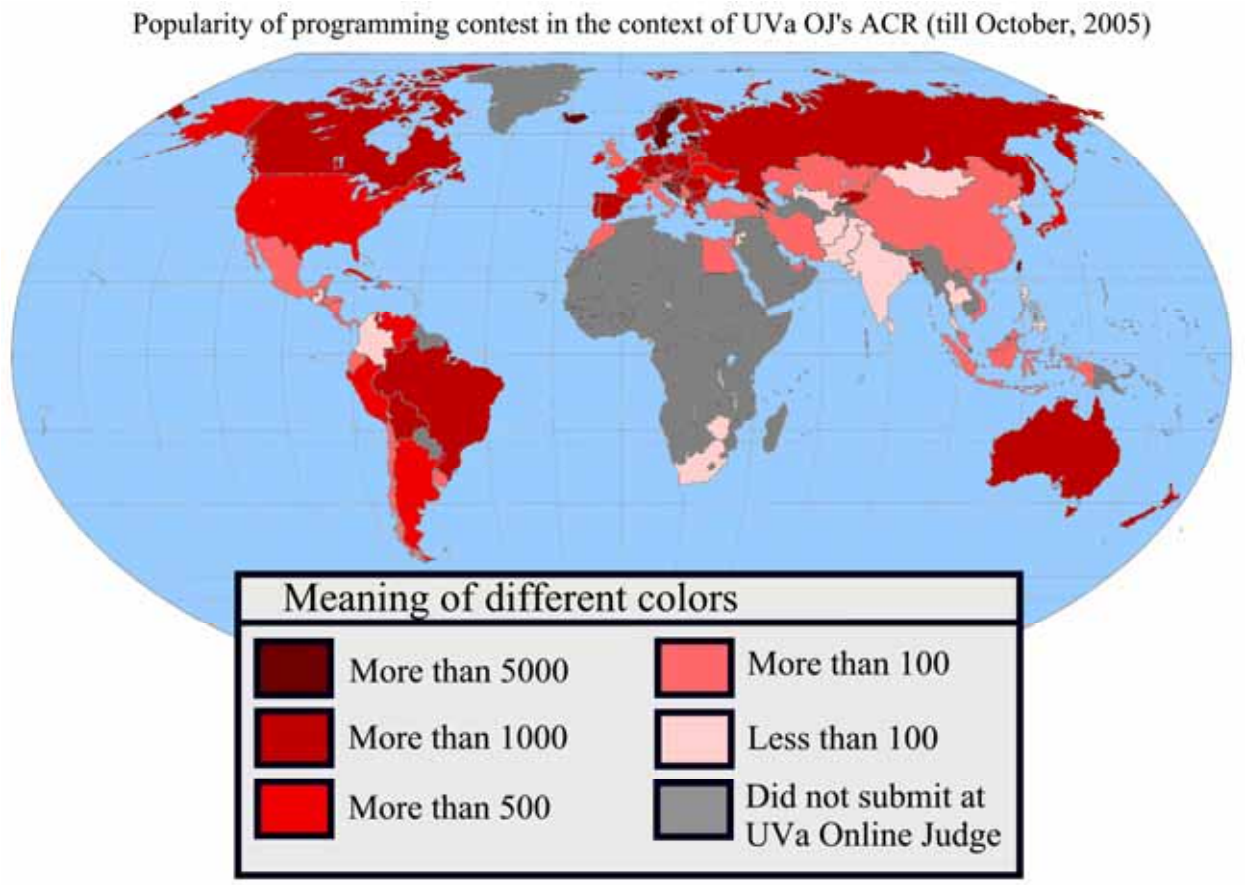
According to this ratio some of the countries where programming contest is most popular are shown in the table below:

Table 12: Countries with high ACR value

Rank	Country	Ratio	Rank	Country	Ratio
1	Hong Kong	30524	21	Hungary	2404
2	Iceland	29610	22	Latvia	2303
3	Taiwan	27004	23	Slovak Republic	1991
4	Estonia	21193	24	New Zealand	1978
5	Singapore	14272	25	Spain	1870
6	Macedonia	8990	26	Bolivia	1863
7	Slovenia	8843	27	Czech Republic	1691
8	Armenia	7143	28	Australia	1665
9	Croatia	6778	29	Romania	1497
10	Sweden	5362	30	Netherlands	1253
11	Portugal	4544	31	Brazil	1170
12	Poland	4515	32	Finland	1084
13	Switzerland	4230	33	Germany	1050
14	South Korea	4103	34	Russian Federation	1037
15	Bosnia-Herzegovina	3797	35	United States	945
16	Bangladesh	3553	36	Venezuela	892
17	Norway	3316	37	Greece	869

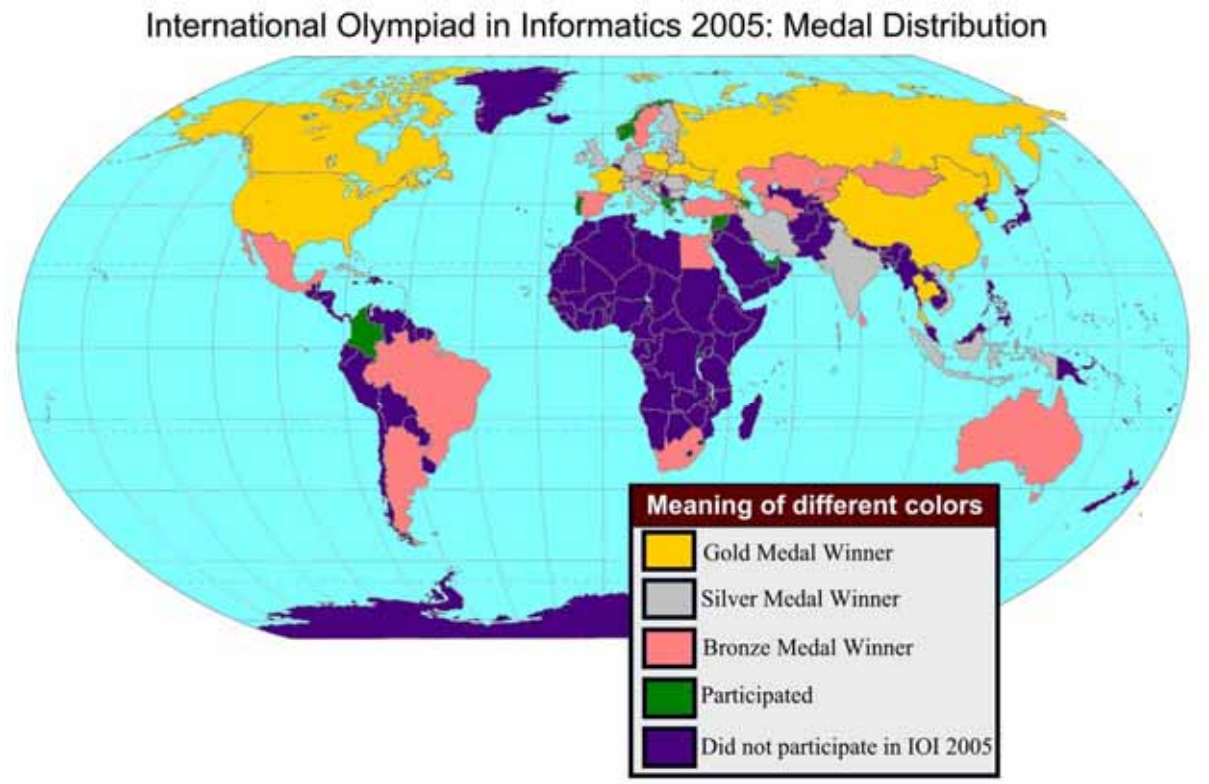
18	Bulgaria	3185	38	Cuba	698
19	Kyrgyz Republic	3155	39	Lithuania	662
20	Canada	2961	40	Belarus	660

The above list is long and it makes us feel good but there are many countries whose citizens have never submitted to Valladolid Site. Some countries who have never submitted to Valladolid Site but has more than 5 million population are (The number in brackets is the population of that country in million): Haiti (8.1), Cambodia, Kingdom of (13.1), Rwanda (8.4), Papua New Guinea (5.7), Mali (13.4), Niger (12.4), Guinea (9.2), Congo, The Democratic Republic of the (58), Syria (18.0), Tadjikistan (6.6), Madagascar (17.5), Malawi (11.9), Laos (5.8), Cameroon (16.1), Ethiopia (72.4), Sierra Leone (5.2), Angola (13.3). And it is obvious that the Subcontinent Africa is mostly out of touch of programming contests. We have made a world map that indicates the growth of programming contest in different parts of the world. The map is shown below:



The darker the red the more is the popularity. They grey countries have either less than 10 members in UVa or they have never submitted to UVa. Population of different countries were collected from 2004 World Population Data Sheet [1].

We have already mentioned before that it won't be appropriate to judge the popularity of programming contest with respect to UVa Online Judge only. So below I represent the maps based on results of IOI and ACM ICPC.



/*All calculations are done manually for IOI. So if you find any mistake please inform the paper author via email*/

In the map above a country is colored yellow if in IOI 2005 the best medal obtained by one or more of its contestants is gold medal, a country is colored silver if the best medal obtained by one or more of its contestants is silver medal. The same rule applies for the countries that have won only bronze medals. The green countries participated in IOI 2005 but did not win any medal. The violate countries did not participate in IOI 2005. Of course there are some countries that are the member of IOI but could not participate because of not getting Visa or other reasons.

Table 13: Best position of countries in ICPC in last eight years (The IBM Years)

Country	2005	2004	2003	2002	2001	2000	1999	1998	Average
Russia	2	1	2	6	1	1	3	2	1.4
Canada	4	12	21	3	4	3	1	3	2.8
China	1	11	5	1	11	4	11	7	3.6
USA	17	5	13	2	2	9	5	5	3.8
Poland	5	10	1	11	6	22	11	9	6.2
Sweden	7	2	13	11	11	22	100	4	7
Germany	100	27	10	18	5	4	2	29	7.8
Australia	41	15	21	11	11	2	28	6	9
Romania	10	100	100	18	48	15	4	10	11.4
Korea	13	15	13	11	8	100	18	100	12
Czech repub	100	27	100	11	14	10	100	1	12.6

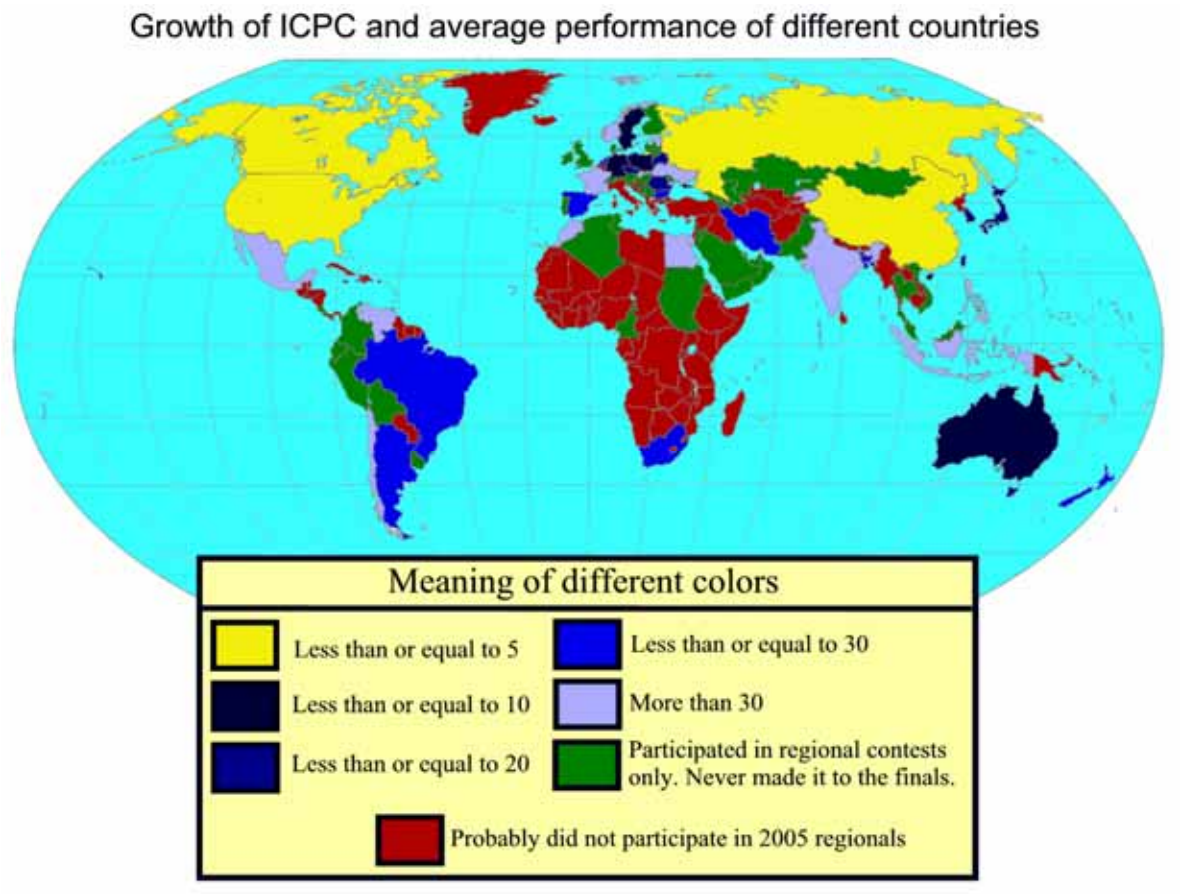
Japan	29	27	11	18	14	7	18	41	13.6
Belarus	17	3	21	18	100	22	28	100	16.2
Taiwan	41	6	56	41	14	39	10	17	17.2
Singapore	29	27	13	41	29	22	18	11	18.2
Hongkong	12	44	30	27	29	8	100	100	21.2
Southafrica	41	15	13	27	29	22	100	100	21.2
Iran	17	27	56	18	9	39	100	100	22
Argentina	41	44	12	10	48	22	28	100	22.6
Bangladesh	29	27	56	41	29	11	39	24	24
Newzealand	41	27	43	27	29	39	11	29	24.6
Netherlands	100	100	30	100	29	22	18	100	24.75
Brazil	29	44	30	41	14	100	28	24	25
Spain	41	15	100	100	29	15	100	100	25
Bulgaria	29	44	100	27	100	100	100	17	29.25
Egypt	41	44	30	41	29	39	100	100	36
India	29	44	43	41	29	39	100	100	36.2
Slovak Republic	100	100	4	100	100	100	18	29	37.75
Mexico	41	44	56	41	48	39	39	29	37.8
Venezuela	41	44	100	100	100	39	100	100	56
Norway	8	27	100	100	100	100	100	100	58.75
Estonia	100	27	100	100	48	100	100	100	68.75
Philipines	41	100	100	100	100	100	100	39	70
France	41	100	100	41	100	100	100	100	70.5
Ukrain	100	100	9	100	100	100	100	100	77.25
Kyrgyzistan	15	100	100	100	100	100	100	100	78.75
Indonesia	100	100	100	100	100	100	100	29	82.25
Macedonia	100	100	100	100	100	39	100	100	84.75
Morocco	100	100	100	100	100	100	39	100	84.75
Chilie	100	100	56	100	100	100	100	100	89

/*All calculations in this table have been done manually (By inspecting each ICPC World Finals Rank list manually). So it is very much possible that this table have some errors. Readers are requested to report about any error to the author via email. In the map below it is very much possible that some of the countries which actually have participated in regional contests have been omitted. Because it is virtually impossible to manually inspect all regional contest rank lists and find out the country of all universities correctly. All data are collected from ICPC Webpage http://icpc.baylor.edu/ */

In the above table the average is done by taking the best five performance of a country in the last eight years. For the countries that have not made it to ACM ICPC World Finals five times, best four performances have been counted. Unlike IOI, ICPC is a university based contest. So it is very difficult for smaller countries to send a team every year to the World Finals. Large countries like Russia, USA, China, Canada do not have such problems because they have many universities that are capable of qualifying and they also have regional contests dedicated for them. The year in which a country has failed to

qualify for the World Finals is given a rank 100 in the table above. The teams getting Honourable Mention in the world finals has been given the next possible rank following the rules for the ranked teams. For example in World Finals 2005 the rank of the last ranked teams is 29 and there 12 teams are ranked 29. So these 12 teams would have got place (29-40) if penalty were calculated. So all the teams getting “Honorable Mention” in this World Finals have been given rank 41. The countries that have never made it to ACM ICPC World Finals have been omitted from the above list. So in last eight years teams from only forty countries have made it to the world finals.

That map below is produced based on the average calculated in the above table. For example the countries which have a position average less than 5 in ICPC are colored yellow, the countries which have position average between 5 and 10 (inclusive) are colored dark blue and so on. The green colored countries have participated in regional contests of 2005 but probably have never made it to World Finals before. The red colored countries probably have not participated in ACM ICPC regional contests of 2005 and also have not made it to world finals before.



Comments and suggestions based on this statistics:
From the statistics above one difference between and IOI and ICPC is evident other than the difference of educational background of the participating contestants.

a) IOI is a nation based contest were almost all nations can send a team of maximum four contestants by fulfilling some requirements on the other hand ICPC is a university based contest where from an university at most one team can qualify for the World Finals. But IOI and ICPC are in two extremes in their rule. IOI ensures the participation of each country but allows only four contestants from all country which may not be enough for large countries like China, Russia or USA. For some of these countries getting into the IOI

team is much harder than winning an IOI gold medal. So this approach may be discouraging for many programmers to be involved with IOI in large countries.

ACM ICPC on the other hand has its problems in their strategy too as it does not put any emphasis on countries officially. For example, just look at the statistics of Slovak Republic or Czech Republic in the above table. They have not made it regularly in the world finals but whenever they have made it they have performed very well. Commonsense suggests that they also had good teams on the years they could not qualify for World Finals but they just did not make it because of very strong neighboring countries. Moreover ICPC is more uncertain in nature than IOI, a good team can accidentally perform badly in the world finals, but in IOI almost all the participating countries get a medal. As every countries don't have a regional contest sites so many don't have chance the participate in ICPC by going into another country. It is not necessary that each country have separate problemset for their regional contests and it is logical to have lesser problemset (Say 15) for all the regional contests combined. This will ensure that a quality problemset is there for all the regional contests. ICPC has some plus points such as their regional contests are more an international event than a national event and hence even more prestigious.

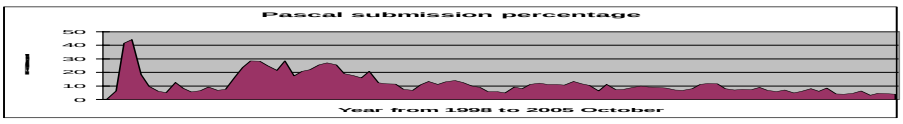
b) The IOI is more certain that ICPC because (i) It allows partial marking unlike the 0/1 approach of ICPC. (ii) It virtually has four teams from a single country so it is unlikely that all of them will do badly. (iii) It requires the contestants to solve only three problems in five hours which is a lot of time. So anyone with a bad start can make up and as there is no penalty on submission time so speed of a contestant is not a strong factor.

Although the grading of ICPC is very strict either correct or incorrect, still it has some very good sides: it gives realtime feedback to contestants about the correctness of their solution and also it is not bad to give some credit to the contestants for their speed.

So to eliminate the short comings these two major types of contests we need a contest that (a) Gives partial marks to contestants (b) Gives real time responses to contestants. (c) Possibly informs the contestant which test cases matches (only the serial of test case) and which don't. (d) If we don't use separate files for each set of input no information regarding correctness will be available if the submitted program does not run within time limit (TL) or crashes (RE) for any one of the inputs. In continuation to this discussion a new probable approach will be proposed after we see some interesting statistics related to programming contest.

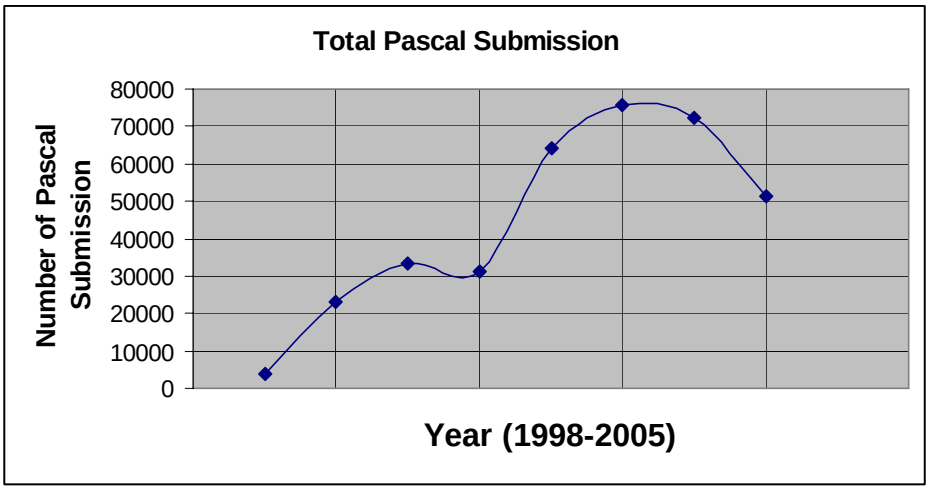
6. Is PASCAL losing its popularity among contestants?

The graph on the right shows how much of the total submission is actually pascal submission in UVa Online Judge from start of 1998 to October 2005. The graph shows that the submission percentage in PASCAL language changes abruptly with time but it also shows that submission percentage



is decreasing. In earlier stages (in 1998) the cause of the abrupt change is something unusual, but we don't want to go into details here. In judging the popularity of Pascal we have discarded all submissions before 2001.

The percentage of submission is decreasing but is the number of submission decreasing?



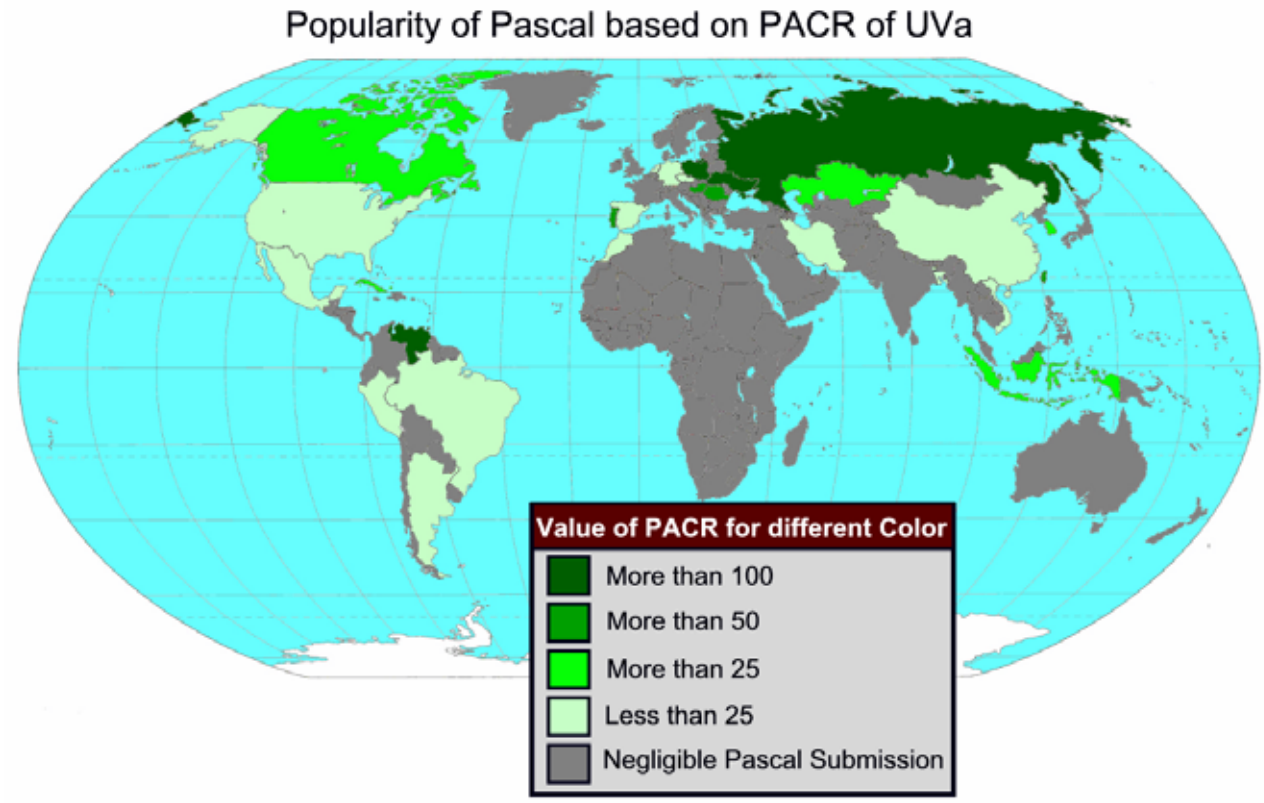
The graph above shows that the number of submission is also decreasing but is it enough to drop Pascal from different contests? Is this drop in submission indicates that lesser people like Pascal or because Pascal is not being used in different programming contests remains so people are being discouraged to use it? All these remain in front of us as food for thought. We are also producing a map that shows the regions where Pascal is still popular. We will use a new ratio PACR (Pascal acceptance ratio) similar to the ratio ACR used in the previous section:

$$\text{PACR}=\frac{Number\ of\ \ Pascal\ submissions\ \ from\ country\ C}{Population\ of\ \ Country\ C\ in\ million}$$

Table 14: Countries with high ACR value

Rank	Country	PACR	Rank	Country	PACR
1	Poland	451.57	15	Mexico	22.89
2	Ukraine	134.09	16	Brazil	22.12
3	Russian Federation	131.83	17	Vietnam	19.67
4	Venezuela	113.78	18	Czech Republic	18.73
5	Portugal	82.29	19	Spain	13.86
6	Hungary	80.20	20	China	11.39
7	Romania	74.93	21	Iran	9.69
8	Taiwan	74.38	22	Netherlands	9.39
9	Kazakhstan	47.20	23	Peru	8.95
10	Cuba	38.67	24	Morocco	4.84
11	Indonesia	32.74	25	Argentina	4.35
12	South Korea	30.15	26	Bangladesh	1.95
13	Canada	25.55	27	United States	1.68
14	Germany	23.41	28		

No matter how optimistic Pascal lovers are it can really be said from this observation that compared to C the popularity of Pascal is approaching zero. Because when one will compare the PACR values with the ACR values of previous section he will find that there is not comparison after all.



7. What type of errors do contestants make most frequently?:

Suppose one submits a program in a contest and gets accepted, another contestant submits a program he gets wrong answer and then he submits again he gets accepted, another contestant submits a program six times and every time he gets wrong answer. Which one of these three events is more likely to happen in a programming contest? To find it out we analyzed all the submissions of UVa site and found out which are the most common response sequence for a contest. We actually took a method like digraph, trigraph analysis of a text. First we tried to analyze which submission response is most common for a problem. And the most frequent response sequences are given in the tables below:

Table 15: A table for most popular response sequence

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

Table 16: A table for most popular responses ending with an AC or having length 6

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

Comments and Suggestions:
Many comments can be made based on these tables. But some things are obvious

a) When a contestant make mistakes for a problem he tends to make the same mistake again.
b) 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.

All these results can help coaches to identify the mistake pattern of his team, one can judge whether a programming contest took place ideally (Must have similarity with these results), in future we may be able to predict the result of a contest based on this. Based on these statistics we might be able to find a correlation between one ideal and another arbitrary contest and this correlation will help us to find how good that arbitrary contest was. But due to the lack of data from realtime contests such statistics is not shown here.

8. The most favorite coding time:

In this section I will show at which part of their day people generally submit more to Valladolid Online Judge. Of course it is not safe to assume that this submission time is always the coding time because many tend to submit from the university whereas the actual coding was done at home. We divided the day into eight 3-hour time slots starting from 0 AM. As it can be seen below that 3 PM to 6 PM slot is the most favorite time slot for coders and the 3 AM to 6 AM slot is the least favorite time slot for coders.

Actual Data

Time Slot	Percentage
0 AM – 3 AM	11.37%
3 AM – 6 AM	3.9%
6 AM – 9 AM	4.41%
9 AM - 12 PM	12.44%
12 PM - 3 PM	16.59%
3 PM – 6 PM	18.29%
6 PM – 9 PM	15.54%
9 PM – 0 AM	17.44%

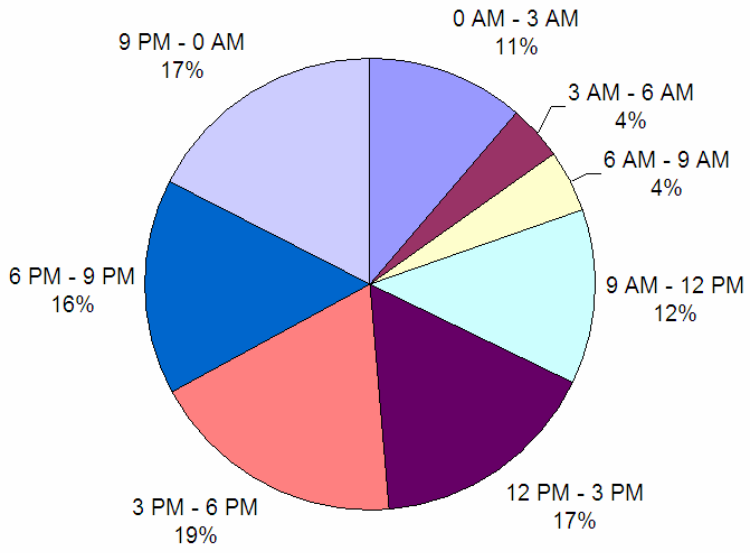


Figure: The submission percentage at different time slot and the corresponding pie chart.

9. The busiest time for the UVa Online Judge:

In the previous section we considered the time zone of the code submitter to find out the most favorite time for submitting codes for users. In this section we are trying to find out the UTC or GMT time when the UVa Online Judge remains the busiest. If someone trying to arrange an online contest for the people all over the world then this section can provide him a guideline to choose the time of his contest. It has been found that from 12 (12 PM)

to 15 (3 PM) GMT the judge remains most busy and from 21 (9PM) to 24 (0 AM) GMT the judge remains least busy.

Actual Data	
Time (GMT)	Percentage
O AM – 3 AM	8.21%
3 AM – 6 AM	11.74%
6 AM – 9 AM	13.70%
9 AM – 12 PM	13.52%
12 PM – 3 PM	16.87%
3 PM – 6 PM	15.99%
6 PM – 9 PM	11.84%
9 PM – 0 AM	8.14%

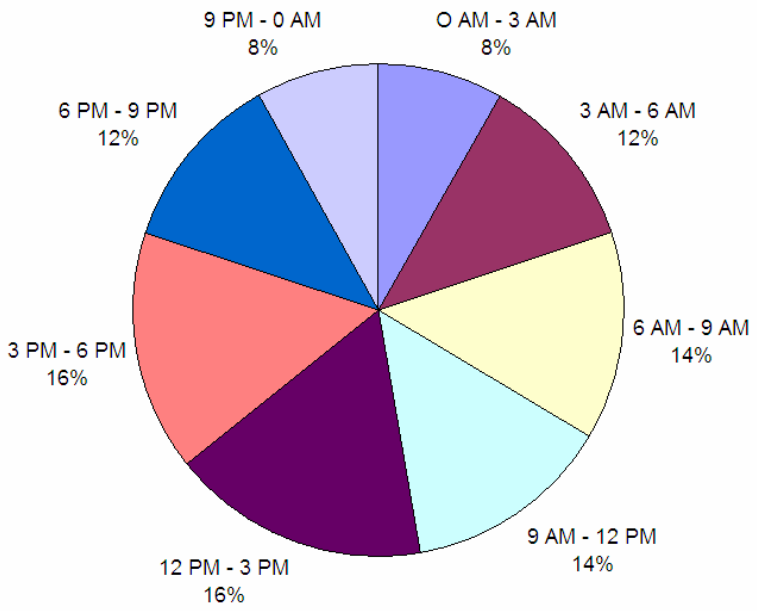


Figure: Submission at different time slot at UVa Online Judge.

10. Acceptance-rejection ratio for Accepted problems only:

Whether it is Online Contest or in 24 Hour Online Judge the acceptance rate is around 30%. But this acceptance rate is not so bad when we consider the statistics of accepted problems only. In this section we will deal only with this submission statistics of accepted problems. For example suppose there are eight problems in a contest A, B, C, D, E, F, G and H. One team solves problem A, B and G and attempts problem C and D. In this section we will not consider the judge responses for problem C and D for that team.

Here we have also divided the judge responses into two types a) Informed Response and b) Uninformed Response. These divisions well help us to propose a combined system to bring IOI and ICPC closer later on. Informed responses are the responses that allow the contestants to know whether their program logic is correct or not correct: AC, PE and WA are such types of responses. The other three TL, RE and CE are uninformed responses, because it is not known what would have happened if the program was allowed to run longer or not crashed. Unless we give one test case per file as input it would be impossible to judge the correctness of the submissions that get TL, RE and CE in present ICPC system. The table below shows the judge response statistics, considering only the submissions from a team for which they finally got an accepted verdict.

Table 17: Judge response statistics for accepted problems/team only.

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

So based on the 135 online contests of UVa Online Judge it is found that given a team will finally get a problem accepted, its probability of getting it accepted in the first submission is 44.16%. The percentage of informed response is 80.89 % and uninformed response is 18.14 %. But more important is the fact that percentage of informed errors is 36.73% and uninformed errors is 18.14%. So their ratio is roughly 2:1.

11. Contestants!! Our Robert Bruces!!!:

"22 mad people are trying to get a half pound ball through the goal post of the opponent" – this can be a funny description of the world's most popular game football. Similarly, ICPC style programming contests can be described as "Three stooges are trying to win a trophy by showing their skill in black box testing". While football is an enjoyable game, so is the programming contest for the contestants. However in the prize giving ceremony the Chief Judge (aka Head Jury) often loves to say how a team failed to solve a problem after submitting it 30 (thirty) times, or another team got a problem accepted in their 20th attempt. These types of things are mentioned because they are rare events in a programming contest. The two tables below show the statistics on how many submissions are required to get a problem accepted. The first table is based on the 135 online contests of Valladolid Site and the second table is based on the 24-hour Online Judge. In the first table we can see that in 10 or less submissions almost 98.6% accepted verdicts are found. It means on average in a programming contest only 1.4% of total accepted problems require more than 10 submissions.

In Online Contests

Table 18: 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

In 24 Hour Online Judge

Table 14: 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	57.798897	57.798897	459607	11	97.313449	0.543648	773820
2	73.200383	15.401486	582077	12	97.749197	0.435749	777285
3	81.574807	8.374425	648669	13	98.084969	0.335772	779955
4	86.649237	5.074429	689020	14	98.359874	0.274905	782141
5	89.991738	3.342501	715599	15	98.589884	0.230010	783970
6	92.320007	2.328269	734113	16	98.782419	0.192534	785501
7	93.961767	1.641760	747168	17	98.942256	0.159837	786772
8	95.179977	1.218210	756855	18	99.070025	0.127769	787788

9	**96.086184**	**0.906206**	**764061**	19	99.189243	0.119218	788736
10	**96.769800**	**0.683616**	**769497**	20	99.284441	0.095198	789493

So in current contest system it won't be illogical too put a bar of 12-15 submissions maximum per each problem.

12. Average Acceptance Time:

This section discusses about a very interesting statistics. In the tables below the column headers denote the number of problems solved at the end of the contest and the row headers denote the current status of the team. So the entry (i, j) or the entry at i-th row and j-th column denotes the average time of getting i-th problems accepted for the teams who eventually solve j problems. For example in the table below the number in red is 133.64. This figure actually means that among all the contests of length five hours or less at UVa Online Judge the teams that solved 7 problems at the end of the contest got their fifth problem accepted after 133.64 minute on average.

Table 19: Considering contests of length five hour or less.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	118.14	86.97	65.15	57.68	42.68	36.28	26.66	27.13	26.54	18.00
2	-1	153.74	115.13	94.15	71.53	59.17	48.59	46.79	49.38	22.75
3	-1	-1	175.25	140.05	105.63	85.91	72.31	70.47	65.38	42.25
4	-1	-1	-1	194.43	149.55	118.32	98.74	99.49	85.54	55.75
5	-1	-1	-1	-1	201.27	158.05	**133.64**	130.71	106.23	69.50
6	-1	-1	-1	-1	-1	203.82	170.93	167.03	132.15	108.00
7	-1	-1	-1	-1	-1	-1	225.82	204.06	161.85	130.00
8	-1	-1	-1	-1	-1	-1	-1	261.76	201.54	147.75
9	-1	-1	-1	-1	-1	-1	-1	-1	247.23	198.75
10	-1	-1	-1	-1	-1	-1	-1	-1	-1	244.50

Table 20: Considering contests of length six hours or less.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	126.75	90.82	69.15	59.17	44.96	37.69	27.90	27.19	27.16	22.29
2	-1	161.05	120.56	97.29	75.65	62.07	49.50	47.17	48.74	38.86
3	-1	-1	183.93	146.05	112.37	90.65	75.34	71.66	64.58	58.57
4	-1	-1	-1	204.35	157.59	125.10	102.82	103.41	84.05	73.00
5	-1	-1	-1	-1	211.70	166.17	139.19	136.71	112.32	92.00
6	-1	-1	-1	-1	-1	216.90	180.86	174.49	140.16	128.57
7	-1	-1	-1	-1	-1	-1	241.79	217.81	169.37	157.14
8	-1	-1	-1	-1	-1	-1	-1	275.27	220.84	179.00
9	-1	-1	-1	-1	-1	-1	-1	-1	275.11	231.86
10	-1	-1	-1	-1	-1	-1	-1	-1	-1	268.57

Table 21: Considering contests of length five hours only

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	142.20	106.70	75.94	68.07	49.83	43.81	28.06	23.90	26.30	18.00
2	-1	184.24	134.56	111.58	83.30	71.36	52.92	44.06	52.40	22.75
3	-1	-1	202.03	164.48	121.21	103.27	78.96	67.08	69.40	42.25
4	-1	-1	-1	225.78	169.65	142.92	108.10	95.38	91.10	55.75
5	-1	-1	-1	-1	226.49	187.72	146.02	119.60	109.00	69.50
6	-1	-1	-1	-1	-1	242.46	186.82	155.83	133.20	108.00
7	-1	-1	-1	-1	-1	-1	244.73	192.29	159.60	130.00
8	-1	-1	-1	-1	-1	-1	-1	254.58	198.50	147.75
9	-1	-1	-1	-1	-1	-1	-1	-1	240.70	198.75

10	-1	-1	-1	-1	-1	-1	-1	-1	-1	244.50

13. “20 Minutes for each wrong submission” – Is it high or low?

ICPC style contests put a lot of weight on contestants’ speed and accuracy. That is why if a team correctly solves a problem after M minutes from the start of the contest and they require total N submissions for it then the penalty added for that problem is M+(N-1)*20 minutes. So for each wrong submission for problem they get 20 minutes of penalty points provided the problem is eventually accepted. But is this penalty too high or low? Let’s try to find an answer.

Considering all the submissions of online contests (Having length between 4 to 6 hours) of Valladolid Site it was found that total 24317 accepted verdict was given and to get these accepted verdicts contestants made 29292 incorrect submissions (PE, WA, TL, RE, CE and others combined). The average acceptance time for each problem was around 127.3474 minutes. So due to incorrect submission the total penalty awarded was 29292*20=585840 minutes and total penalty due to submission delay was 127.3474*24317~3096707 minutes. So the total penalty= 3096707+585840 = 3682547 minutes. So on an average the penalty due to submission mistakes or wrong verdicts was $\frac{585840}{3682547} * 100\% = 15.91\%$ of the total penalty. Now it is up to the reader to decide whether it is high or low.

Suppose that we want the penalty due to wrong verdicts be w % of the total penalty on an average, then what will be the value of penalty x per submission. Let’s find out:

$$\frac{29292 * x}{29292 * x + 24317 * 127.3474} * 100 = w$$

$$or,\ 2929200x = 29292wx + 3096707w$$

$$or,\ 2929200x - 29292wx = 3096707w$$

$$or,\ x(2929200 - 29292w) = 3096707w$$

$$or,\ x = \frac{3096707w}{2929200 - 29292w}$$

14. A New Contest Model:

It has already been said that an ideal contest model should have partial credits like IOI and and also realtime feedback like ICPC. But ICPC allows the contestant to submit problem infinite times. But giving partial credit and infinite time submission is a bit to much because in each submission the contestant has the option to try different kinds of tests and morever if he is allowed to know which test cases are getting wrong he might use one of his solution to produce output for some test cases and another solution to produce outputs for other cases just depending on the case number. In our study we also found that the ratio of informed and uninformed errors is roughly 2:1. So we can set a new limit that a team will be allowed to make total eight wrong submissions per problem and another four uninformed responses will be allowed. So a team can get 4 RE and 8 WA for a problem but he cannot get 9 WA because maximum 8 informed errors will be allowed. In other words we can say that total 8 errors will be allowed and first four uninformed errors will not be counted in these eight errors. With this new rule the statistics of Table 14 becomes:

Table 14: 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

As we are allowing 8 errors so if the ninth submission is an accepted verdict, it will be granted. However if a team fails to get the problem accepted in these submissions he will be given the highest point that he obtained among these submissions.

Now the question comes how can we prevent poorly written solutions to get good scores? – in this model the answer is simple. As we are allowing the contestant to fix his mistakes we don't need to be as lenient as the current IOI, so partial marks will only be given if someone gets more than 60% marks, otherwise he will get a zero.

Now the question that may come how will weak coders get marks as there is no lenient rule like the 50% rule [2], and the answer is just to give an easy problem to the contestants to solve so that they can get some marks and let the hard ones remain hard. The total number of problems can also be increased (Say five problems in five hours) to include easy and easy medium problems.

15. Conclusion:

Programming contest has been around for a long time and has been quite successful in creating programmers of the highest quality. It has its critics and it has its limitation in its judging system surely, but still it is better than many widely used systems. It is important to note that the subject of programming contests is human being and so its accuracy cannot be maximized by calculating the derivative of a mere expression. As we are humans and not robots so we are bound to make some mistakes. No matter how long we practice our accuracy will not be more than a certain limit and while making rules we need to take into account our average properties as human problem solvers. That is why the goal of this paper is to present the huge database of Valladolid Online Judge in different ways that may help the decision makers to shape up programming contest in a better way. Yes, the gives the model of a new contest but that is not the only conclusion that can be drawn from these statistics. The paper in inclined towards ICPC because it is based on an ICPC type Online Judge and as the author believes that realtime response is a must ina programming contest. But the model that is good for college students should be good for university students and vice versa, may be the duration and the team size can remain different as it is now. Many new aspects of programming contest has been described by Cormack et. Al. [3] in detail and none of these actually contradicts the ideas presented here.

The problem with an ideal programming contest model it that it needs to fair but it also needs to be simple because the same model will be followed in regional (ICPC) and

national contests (IOI). Also some of the models are extremely popular so it will take some time to replace them and all the online judges are written in the existing rule and it will take some time to change them as well. Many regions and nations are still struggling to adopt the present simple contest models so the new more complex models can be impossible for them to follow. So a new full proof system can first be followed in international level and then in course of time poured into national and regional level.

16. References:

[1] World Population Data Sheet 2004 of the Population Reference Bureau, Link: http://www.prb.org/pdf04/04WorldDataSheet_Eng.pdf

[2] Yakovenko B., 50% rule should be changed.

[3] Cormack G., Kemkes G. , Munro I., Vasiga T., Structure,Scoring and Purpose of Computing Competition

Acknowledgements:

Prof. Miguel A. Revilla:
He is the main responsible person of UVa Online Judge, official archivist of ACM International Collegiate Programming Contest (ICPC), CII (Competitive Infrastructure Initiative) Member. He is a very busy person but still he managed time to supply me the data from UVa site that I requested for.

Prof. Gordon V. Cormack:
For his suggestion to write something based on the statistics of UVa Online Judge. His comment on the ideas presented in this article was, "All of those questions look interesting to me. I'm not sure which to focus on. ***You may notice interesting things once you start to dig.****" The bold part of his suggestion proved very much true while the author was digging through the statistics and at one stage so many new things came up that the author was worried about the completion of this paper.*

Carlos M. Casas Cuadrado:
He is a very important member of Valladolid Online Judge and maintains the problemset section very efficiently.

Derek Kisman:
Derek Kisman is a world class problem setter (ACM ICPC World Finals Judge and former ECNA judge) but he has also helped the author to produce many good and error free problems by writing alternate solutions and providing useful suggestions.

Monirul Hasan:
Before this guy left for Canada for higher studies, he and the author were the main persons to set problems for many programming contests of Bangladesh. He is still a member of Elite Problem setters' Panel of UVa Online Judge.

Special thanks also goes to Prof. William B. Poucher, Prof. M. Kaykobad, Prof. Abul L. Haq, Md. Sajjad Hossain, Miguel Revilla (Junior), Syed Monowar Hossain, Fernano P. Najera Cano, Jimmy Mårdell, Rujia Liu, Md. Kamruzzaman, Igor Naverniouk and Joachim Woulf for their encouragement and support in different programming contests.