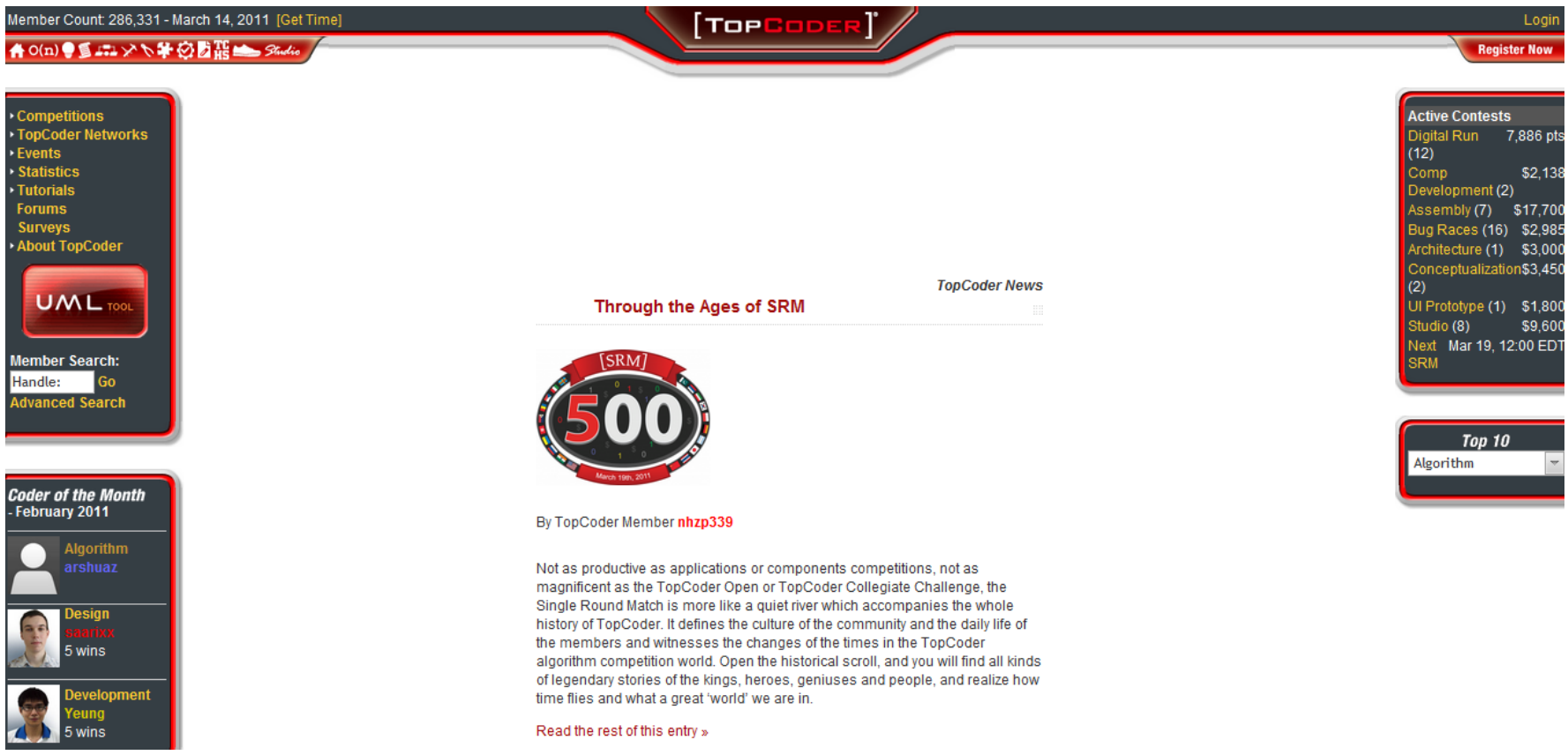
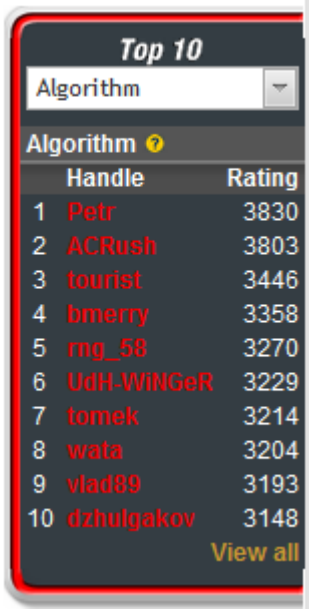
打开 http://www.topcoder.com/tc 是这个样子：

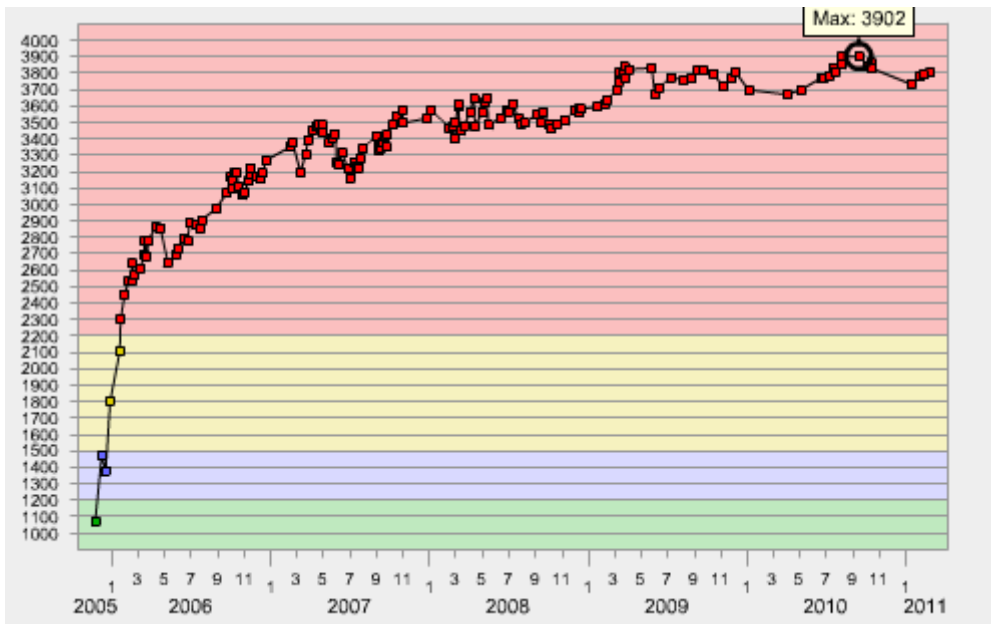


2 右端有一个 TOP 10，点开 Algorithm，就是我们要玩的算法竞赛的 TC 排名 TOP10：



写这文的时候第一名是 petr，第二名是 ACRush。后者就是大名鼎鼎的楼天城，国内公认的算法竞赛第一人。第三名是 tourist，一个白俄罗斯的初中生……第七名 tomek 是曾经的领军人物。

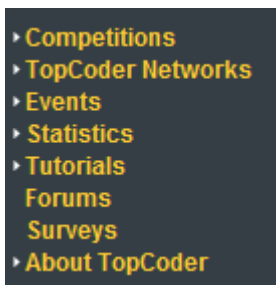
点开人名可以看到更多的信息：



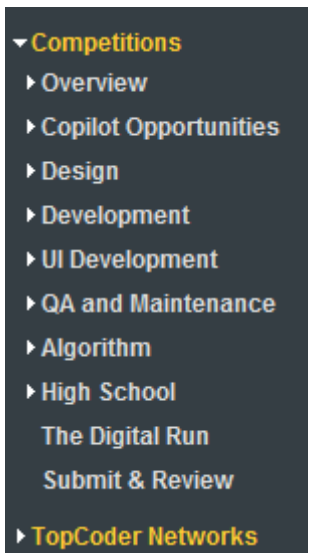
ACRush 的积分走势图

楼天城玩这个已经 5 年有余了，可见此比赛的质量和影响力。积分的英文名叫做 rating。

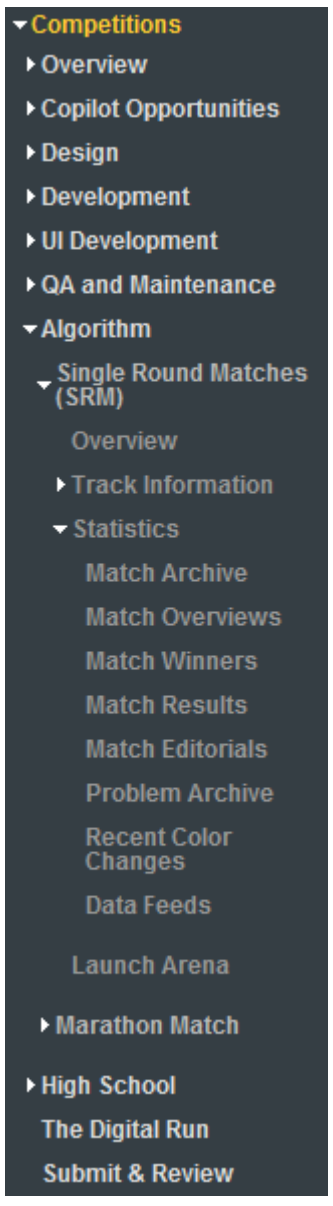
退出人名页面，回到主页，往左看：



先点开 Competitions：

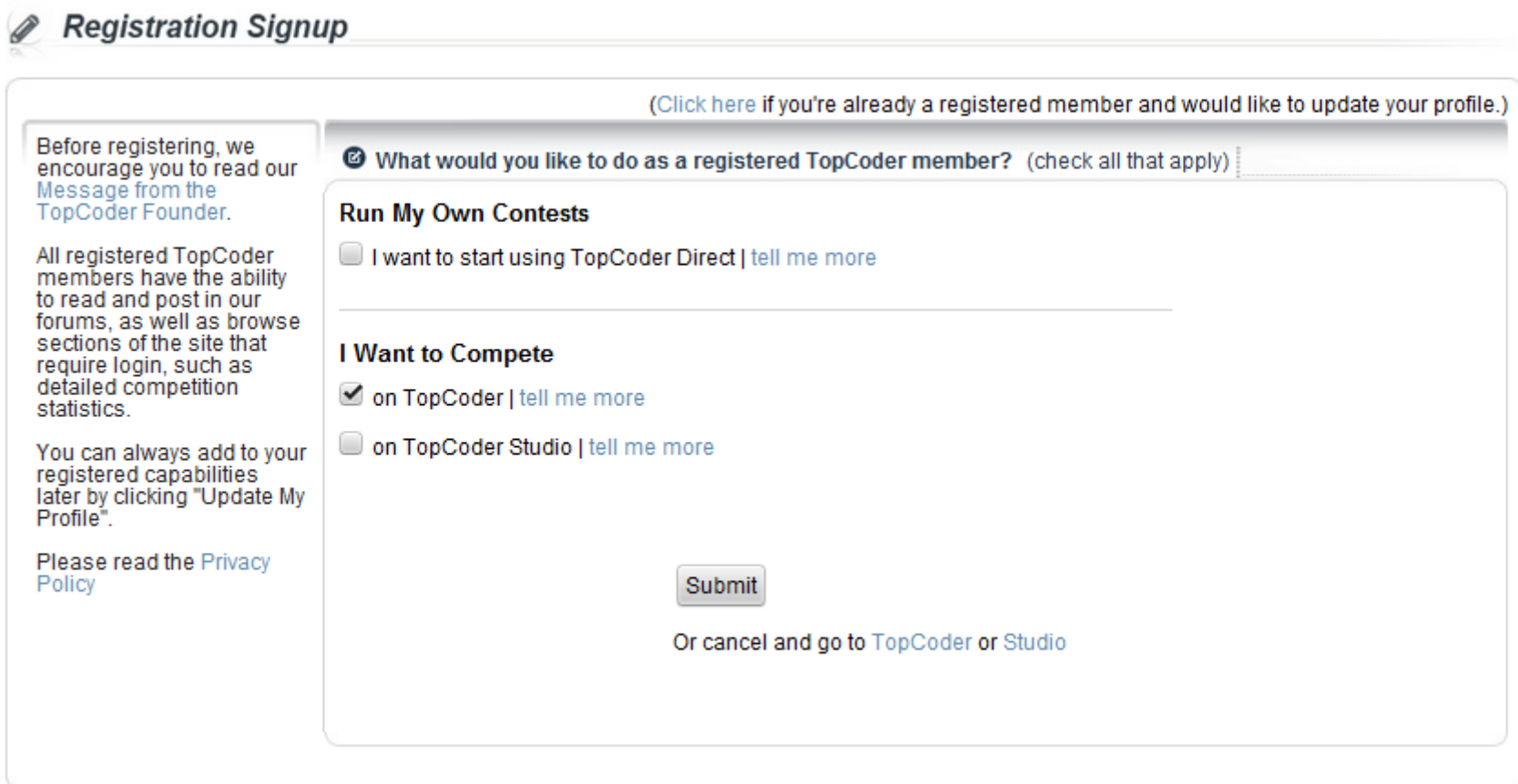


然后是这样：
OVERVIEM 里面有一个 rating 项，点开有 Rating 计算公式的详细介绍。
http://apps.topcoder.com/wiki/display/tc/Component+Development+Ratings



算法类里面里面 SRM 就是我们要玩的比赛。一个月 2-4 场。
Match Editorials 里面有解题报告。其他的就不介绍了，自己探索一下？

然后我们去注册一个账号吧！主页右上角：Register Now。



这样，然后 submit。想了解详情看 tell me more。

* First Name:	Yida Given Name	依次是 名字
* Last Name:	Xie Surname	姓。
* Current Address 1:	Chengdu	地址
Current Address 2:		
Current Address 3:		
* City:	Chengdu	城市
State (US Only):		
Postal Code:		
Province:		
* Country:	China	国籍：中国
* Country to represent:	China	代表国籍：中国
* Timezone:	Asia/Shanghai	时区
Phone Number:		
* Email Address:	Yida@kikoku.com	邮箱
* Confirm Email Address:	Yida@kikoku.com	确认邮箱
Email Notifications:	☑ Algorithm Competitions ☐ Software Development Opportunities ☐ PayPal X Contests and Opportunities powered by TopCoder ☐ Employment Opportunities ☐ TopCoder News & Events	
* Enable Member Contact:	○ Yes ◉ No	
* Show/Hide earnings:	○ Show ◉ Hide	

可以乱填，但是以后比赛拿了钱可能就麻烦了。

Account

* Username:	XieYida	依次为 用户名（你登陆用的）
* Password:	•••••••	密码
* Confirm Password:	•••••••	重复密码
* Secret Question:	SMX=?????? secret question must be at least 8 characters in length.	密码保护问题
* Secret Question Response:	••••••••••	回答
Quote:	All are souls.	签名
* Student/Professional:	Student	教授？学生？

☑ I have read and accept the Terms of Use

Submit

Or cancel and go to TopCoder or Studio

然后 submit。如果哪项没填好，它会提示你。比如说我第一次的时候密保问题长度太短了。

Demographic Information

* Age:	18 - 24	年龄
* Gender:	Decline to Answer	性别（拒绝回答）
* Ethnic Background:	Decline to Answer	种族（拒绝回答）
* Primary Interest in TopCoder:	Technology Competition	主要兴趣
* Shirt Size:	Medium In the event that you win/earn a shirt for a TopCoder event, it is convenient for us to have your shirt size on record. If you choose not to answer this question, we will assume that you do not want any shirts.	T-shirt 型号
* College Major:	Computer Science	专业
* College Major Description:	CS	专业描述
* Degree Program:	Bachelors	学位
* Graduation Year:	2014	毕业年份
* Graduation Month:	July	月份
* Clubs / Organizations:	IEEE ACM SWE Systers Choose all that apply	组织
Other Clubs / Organizations:		
* School:	Choose School	学校，点进去后搜索 University of Electronic Science and Technology of China 加那个人数最多的。
* Show / hide my school:	Show Hide	
GPA:		
GPA Scale:		
Resume:	选择文件 未选择文件	
* How did you hear about TopCoder?:	Member Referral	怎么知道 TC 的？（会员推荐）
* Referring member's handle:	ACRush	哪个会员推荐的？ACRush（XD）

Submit
然后给你一个页面让你确认，确认无误后 submit。

Registration Signup

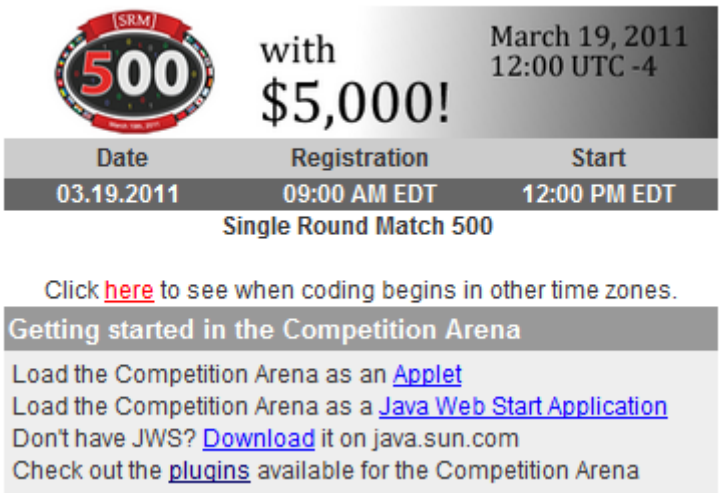
Thank you for registering.
We have sent you a confirmation email with activation information.
Please check your email and activate your account now.

然后去邮箱找到激活邮件激活账号。
参加比赛。主页右边，Events->Events Calendar。

- Competitions
- TopCoder Networks
- Events
  - Event Calendar
  - Tournaments
- Statistics
- Tutorials
- Forums
- Surveys
- About TopCoder

Sun	Mon	Tues	Wed	Thurs	Fri	Sat
		1	2	3	4	5
6	7	8 SRM 499 11:00	9	10	11	12
13	14	15	16	17	18	19 SRM 500 12:00 $5,000
20	21	22	23	24	25	26
27	28	29	30 Member SRM 501 7:00	31		

每场比赛的时间（不是北京时间）。点进去。



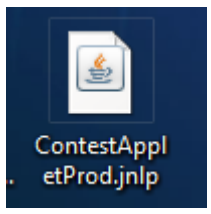
点 here 看比赛在其他时区的开始时间。

Rome	Sat 17:00
San Francisco *	Sat 09:00
San Juan	Sat 12:00
San Salvador	Sat 10:00
Santiago *	Sat 13:00
Santo Domingo	Sat 12:00
São Paulo	Sat 13:00
Seattle *	Sat 09:00
Seoul	Sun 01:00
Shanghai	Sun 00:00
Singapore	Sun 00:00
Sofia	Sat 18:00
St. John's *	Sat 13:30
Stockholm	Sat 17:00
Suva	Sun 04:00

Shanghai 就是北京时间（周日 0 点）。

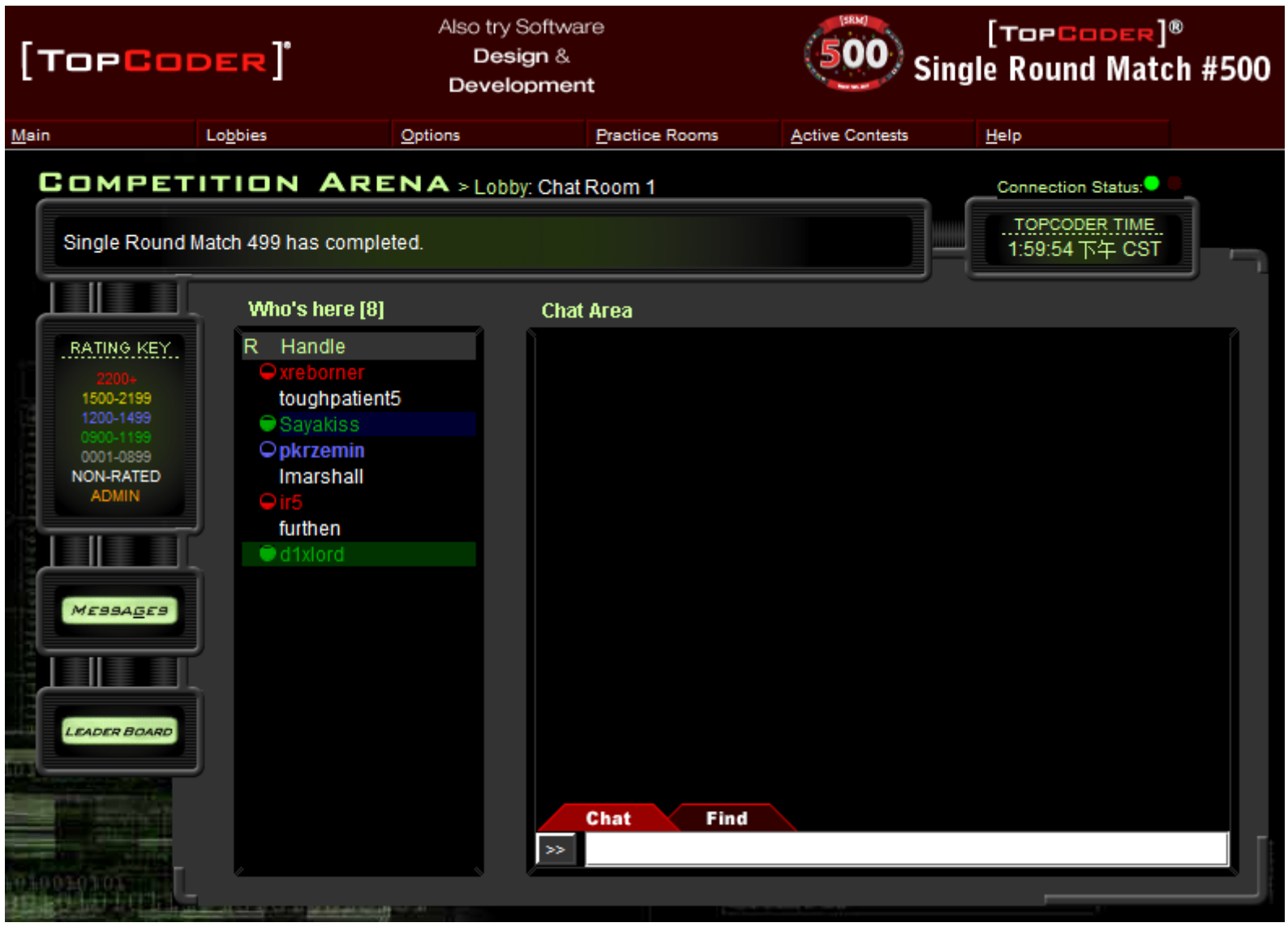
回到主页，点右上角的 O(n):

点 load competition arena。（competition arena 就是比赛客户端，直译为“竞技场”）然后下载。

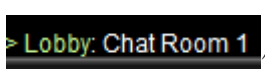


大概这样。双击运行。如果无法运行，请检查是否安装了 JRE 或者 JDK，如果不是这个问题，就到网上或者论坛或者群里面问。

输入账户和密码



进去后是这样。

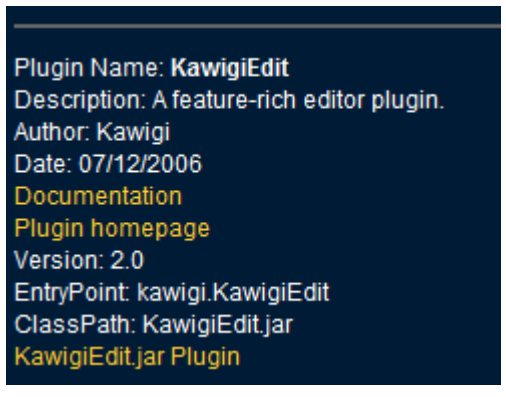


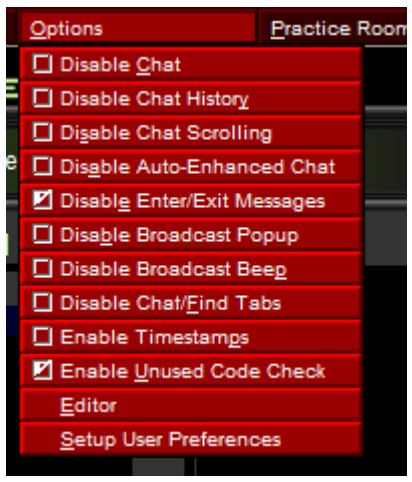
你所在的位置。

链接状况。

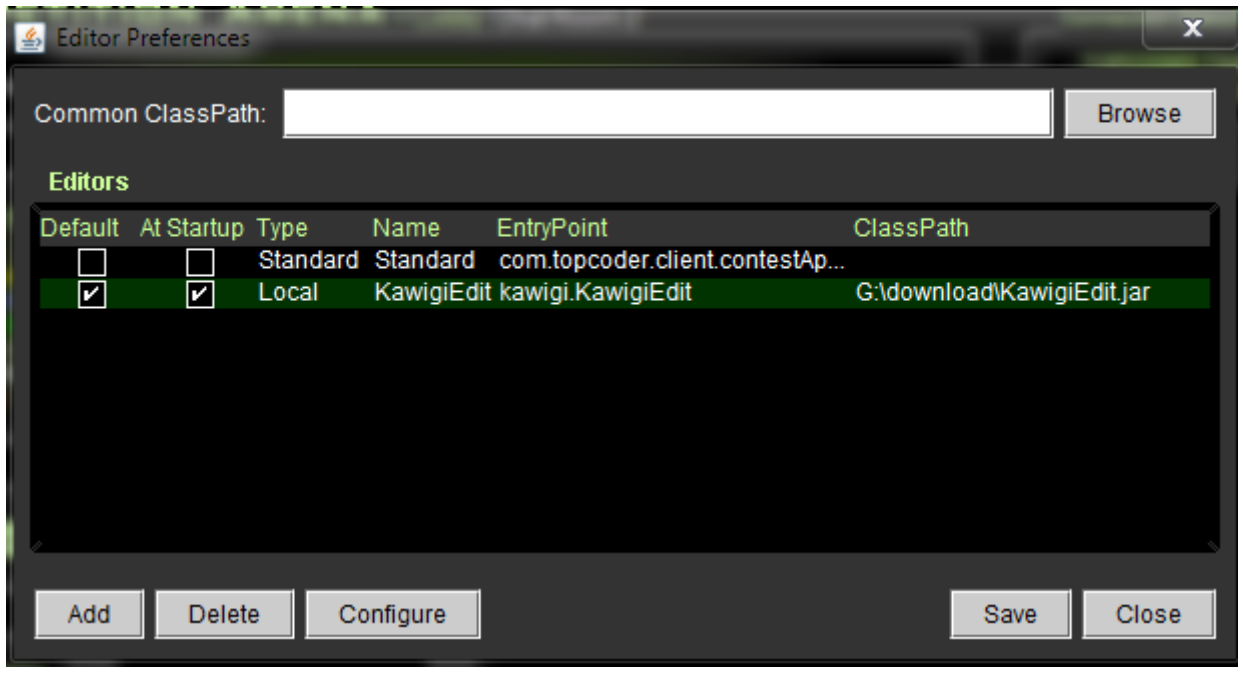
点开 lobbies，可以看到有哪些房间给你进。CHAT ROOM 就是一般聊天用，Admin 那个是给管理员用的。

（非常重要）去这里下载一个很有用的插件 KawigiEdit.jar https://www.topcoder.com/tc?module=Static&d1=applet&d2=plugins



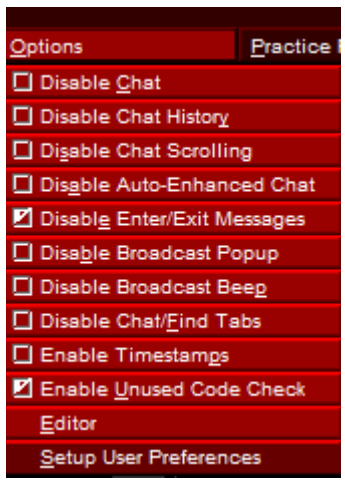


点 EDITOR 修改编辑器。

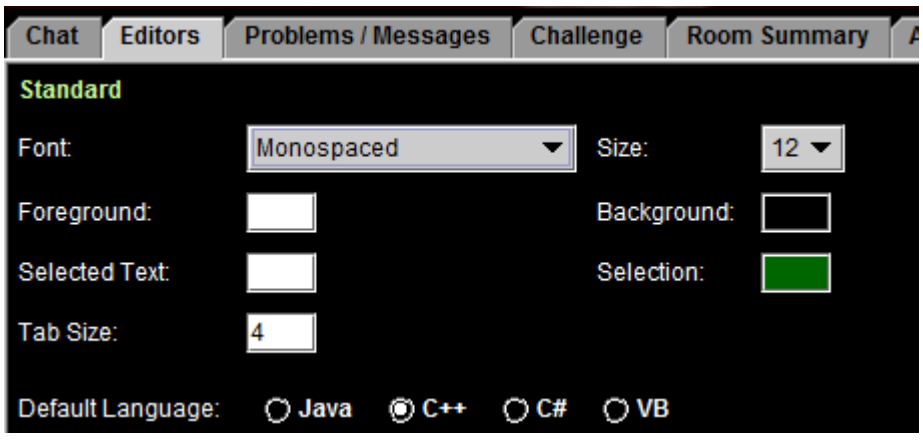


点 ADD 添加编辑器。

用这个编辑器，会让你节约很多调试时间，并且让你快速上手。



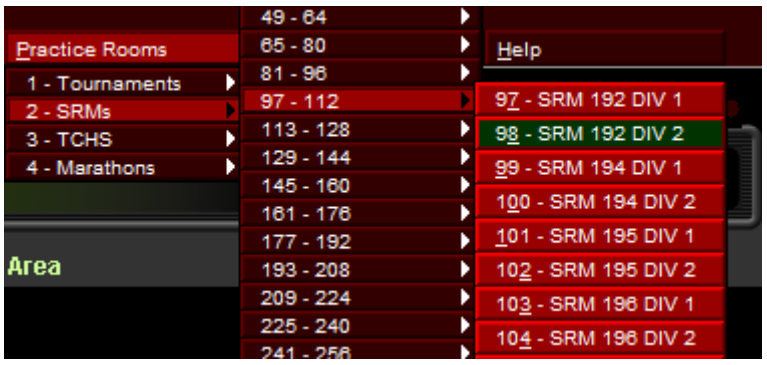
最后一项可以设定字体等等诸多细节。



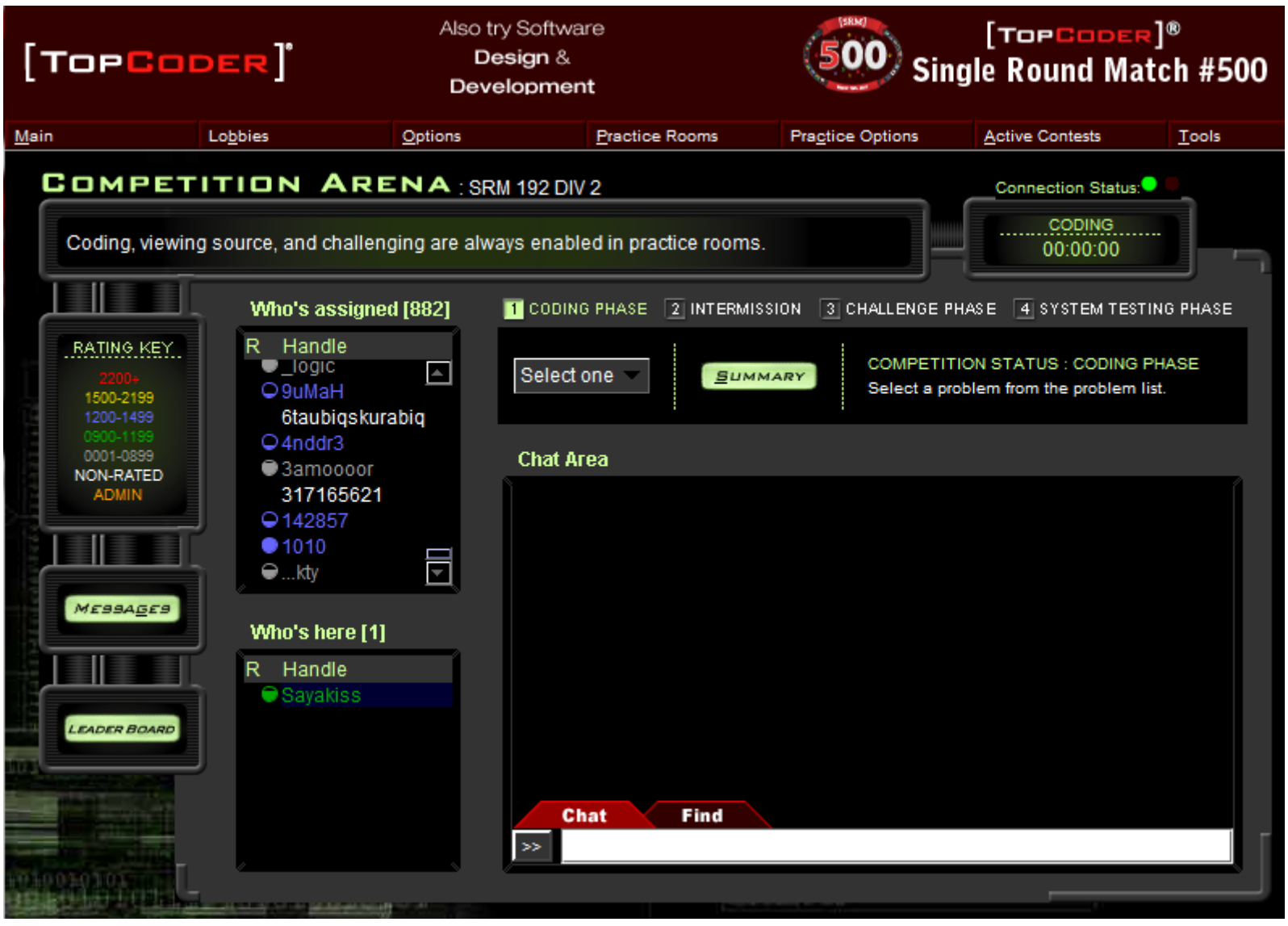
在这里把默认语言设为你常用的那种语言。这样能节省掉几次点击。

点开 Practice Room。

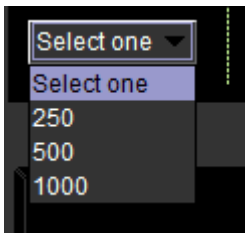
SRM 是我们要玩的。



随便找一场 SRM 来试试吧。DIV1 较 DIV2 来说题目难度要高一个层次。我们就去 SRM192 的 DIV2 吧。



进去后是这样。在 select one 那里开题：



一般来说有 250/500/1000 分三个等级的题。分数越高难度越大。一般来说 DIV2 的 500 分的题=DIV1 的 250 的题。

一般来说 DIV2 的 250 题一般来说是无算法的纯水题。500 分就是有点思维或者暴力题之类的。

我们点开 250 来看看。

上面是题，下面是代码框，左边是这道题还剩多少分数。

分数随着时间流逝而逐渐减少。先减少的慢，然后快，然后慢，然后不减。

我们来看看题目。

Your company has a new box loading robot. It is your job to program it to pack items into the shipping boxes. The robot does not have a very large program memory so you are restricted to placing all the items into the boxes in the same orientation. Each item is a rectangular solid with dimensions **itemX** by **itemY** by **itemZ**. The box is also rectangular with the dimensions **boxX** by **boxY** by **boxZ**. The items can be placed in the box in any orthogonal orientation (ie. the sides of the items must be parallel to the sides of the box), but only whole items can be placed in the box. Your task here is to determine the greatest number of items that can be packed into the box (with all the items in the same orientation).
For example, if the box is 100x98x81 units and the items are 3x5x7 units, then orienting the items so they are 5x7x3, allows them to fit in the box in a 20x14x27 grid, filling the entire box, which is optimal: 7560 items.

看不懂的题的话，及时查词典。这也是练你的英语阅读。

Definition

Class:	BoxLoader
Method:	mostItems
Parameters:	int, int, int, int, int, int
Returns:	int
Method signature:	int mostItems(int boxX, int boxY, int boxZ, int itemX, int itemY, int itemZ)
	(be sure your method is public)

因为 SRM 的比赛程序是要求编写一个类来解决问题（面向对象的思想），所以你需要把你的类写来符合题目要求。但是有插件的话，它可以自动帮你生成这个类：

```
#include <sstream>
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <cmath>
#include <cstdlib>
#include <ctime>

using namespace std;

class BoxLoader {
public:
    int mostItems(int, int, int, int, int, int);
};

int BoxLoader::mostItems(int boxX, int boxY, int boxZ, int itemX, int itemY, int itemZ) {

}
```

还把各种 include 帮你写好了。好用吧？

Constraints

- **boxX**, **boxY** and **boxZ** will be between 1 and 1000 inclusive.
- **itemX**, **itemY** and **itemZ** will be between 1 and 1000 inclusive.

数据范围

```
687

783

109

93

53

Returns: 833
```

样例

代码可以直接在 EDITOR 里面编写，也可以复制到 IDE 里面编写，我一般选择复制出来。

```
stItems(int boxX, int boxY, int boxZ, int itemX, int itemY, int itemZ)
```

这里就是题目的输入，测试时系统会把这些数据输入到你的类里面。

```
int BoxLoader::mostItems(int nx, int ny, int nz, int mx, int my, int mz) {
```

输入可以简写成这样，节约时间。

有时候输入数据的类型是 string，vector 之类的，所以需要熟悉基本的标准库类型类型。具体自行搜索。

```
int BoxLoader::mostItems(int nx, int ny, int nz, int mx, int my, int mz)
{
    int xx=nx/mx;
    int xy=nx/my;
    int xz=nx/mz;
    int yx=ny/mx;
    int yy=ny/my;
    int yz=nz/mz;
    int zx=nz/mx;
    int zy=nz/my;
    int zz=nz/mz;
    int ans=0;
    ans=max(ans,xx*yy*zz);
    ans=max(ans,xx*yz*zy);
    ans=max(ans,xy*yz*zx);
    ans=max(ans,xy*yx*zz);
    ans=max(ans,xz*yy*zx);
    ans=max(ans,xz*yz*zy);
    return ans;
}
```

代码写好了。因为这是一个类，所以记得把要求你返回的东西 return 回去。有时候是一个整数，有时候也是一个 vector 之类的。

这里是调试代码，可以直接测试样例：

Source Code | Test Code | Compile | Output | Local Code

```
<%:start-tests%>
double test0() {
    int p0 = 100;
    int p1 = 98;
    int p2 = 81;
    int p3 = 3;
    int p4 = 5;
    int p5 = 7;
    BoxLoader * obj = new BoxLoader();
    clock_t start = clock();
    int my_answer = obj->mostItems(p0, p1, p2, p3, p4, p5);
    clock_t end = clock();
    delete obj;
    cout <<"Time: " <<(double)(end-start)/CLOCKS_PER_SEC <<" seconds" <<endl;
    int p6 = 7560;
    cout <<"Desired answer: " <<endl;
```

调试程序可以把调试代码复制出来到你的代码的后面进行调试：

```
    int ans=0;
    ans=max(ans,xx*yy*zz);
    ans=max(ans,xx*yz*zy);
    ans=max(ans,xy*yz*zx);
    ans=max(ans,xy*yx*zz);
    ans=max(ans,xz*yy*zx);
    ans=max(ans,xz*yz*zy);
    return ans;
}

<%:start-tests%>
double test0() {
    int p0 = 100;
    int p1 = 98;
    int p2 = 81;
    int p3 = 3;
    int p4 = 5;
    int p5 = 7;
    BoxLoader * obj = new BoxLoader(
    clock_t start = clock();
```

`<%:start-tests%>` 形如这样（<%XXXX%>）的语句把他删掉。否则无法同过编译。

```
    time = test3();
    if (time < 0)
        errors = true;

    time = test4();
    if (time < 0)
        errors = true;

    if (!errors)
        cout <<"You're a stud (at least on the example 
    else
        cout <<"Some of the test cases had errors." <<e
        system("pause");
}

//Powered by [KawigiEdit] 2.0!
```

在程序的最后加上 system("pause");
这样才能看到输出结果。
输出：

```
Time: 0 seconds
Desired answer:
        7560
Your answer:
        7182
DOESN'T MATCH!!!!

Time: 0 seconds
Desired answer:
        0
Your answer:
        0
Match :-)

Time: 0 seconds
Desired answer:
        100
Your answer:
        1500
DOESN'T MATCH!!!!

Time: 0 seconds
Desired answer:
        833
Your answer:
        1904
DOESN'T MATCH!!!!

Time: 0 seconds
Desired answer:
        20
Your answer:
        48
DOESN'T MATCH!!!!

Some of the test cases had errors.
请按任意键继续. . .
```

挂了不少组，回去看看是哪写错了。

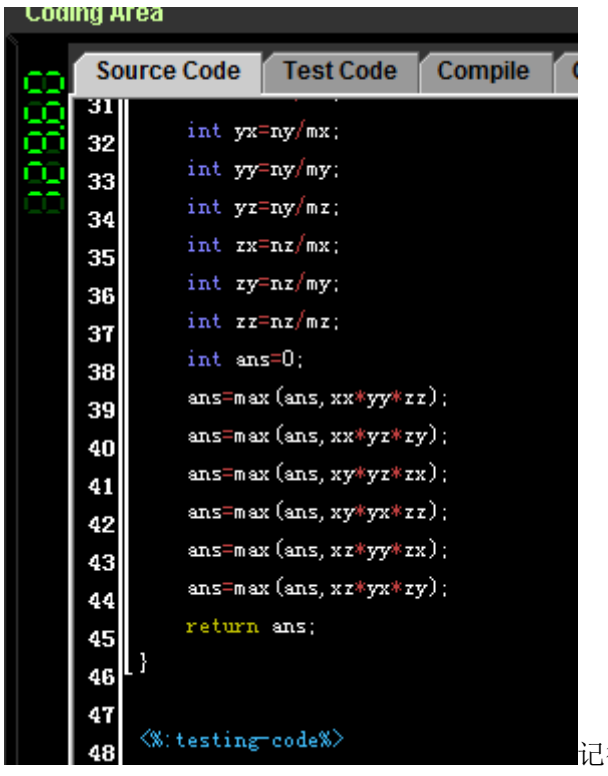
```
int xx=nx/mx;
int xy=nx/my;
int xz=nx/mz;
int yx=ny/mx;
int yy=ny/my;
int yz=ny/mz;
int zx=nz/mx;
int zy=nz/my;
int zz=nz/mz;
int ans=0;
ans=max(ans,xx*yy*zz);
ans=max(ans,xx*yz*zy);
ans=max(ans,xy*yz*zx);
ans=max(ans,xy*yx*zz);
ans=max(ans,xz*yy*zx);
ans=max(ans,xz*yx*zy);
return ans;
```

几个小 bug，改成这样：

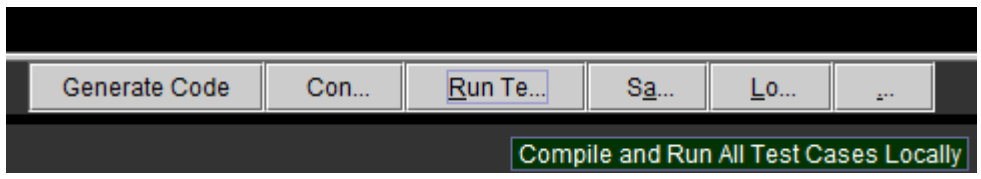
```
Time: 0 seconds
Desired answer:
        7560
Your answer:
        7560
Match :-)

Time: 0 seconds
Desired answer:
        0
Your answer:
        0
Match :-)

Time: 0 seconds
Desired answer:
        100
Your answer:
        100
Match :-)

Time: 0 seconds
Desired answer:
        833
Your answer:
        833
Match :-)

Time: 0 seconds
Desired answer:
        20
Your answer:
        20
Match :-)

You're a stud (at least on the example cases)!
请按任意键继续. . .
```

过样例。

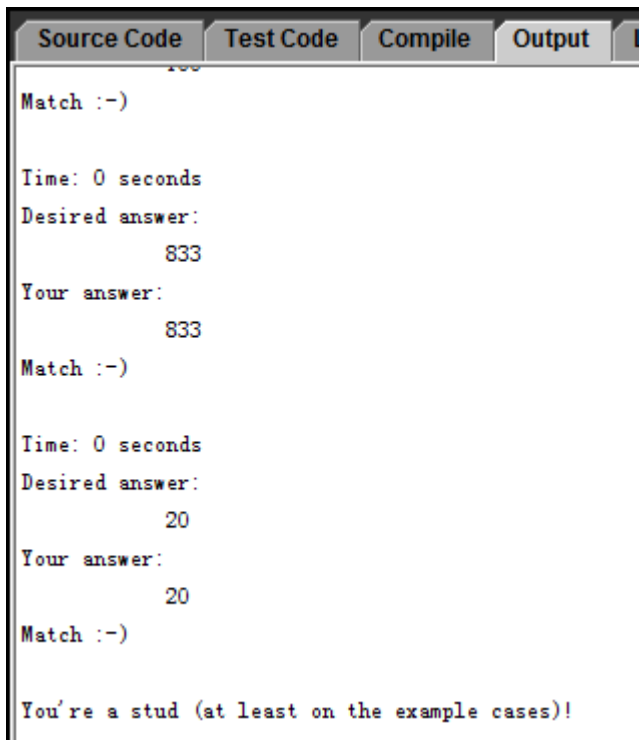
调试也可以把代码复制到 Source Code 里面去，然后利用 Editor 直接调试：



记得保留末尾的<%XXXX%>，否则无法直接调试哟。



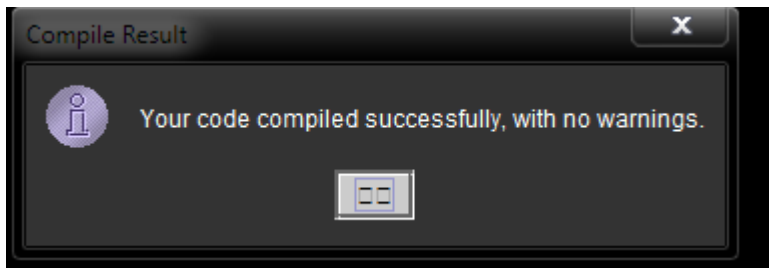
点 RUN TEST。



返回结果。

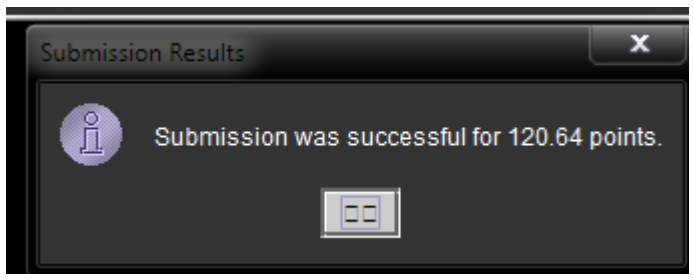
好了，过了样例之后，感觉没问题就可以交题了!、

点右下角的 COMPILE（那个 TEST 是 ARENA 自带的测试工具，一次只能测一组样例）



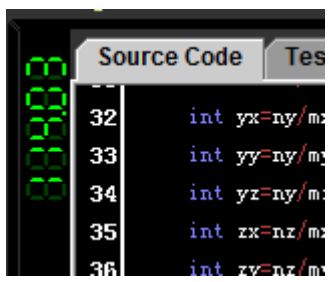
没问题!（当然了，因为我的代码是在本地 IDE 里面成功编译过的）

然后点右下角的 SUBMIT 交题!

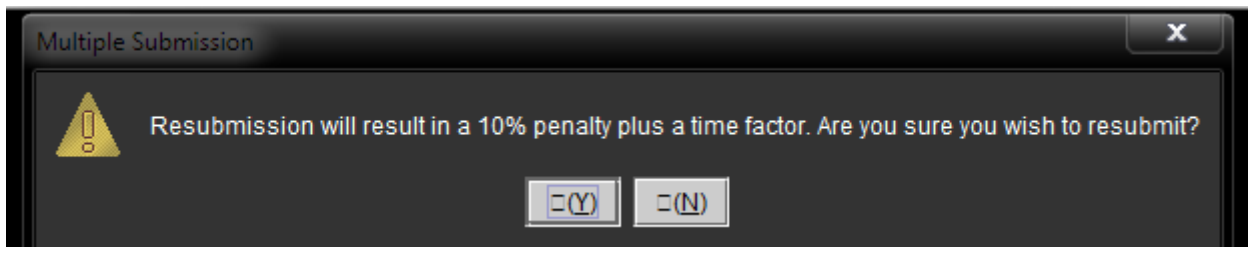


告诉你这题你得到了 120.64 分。

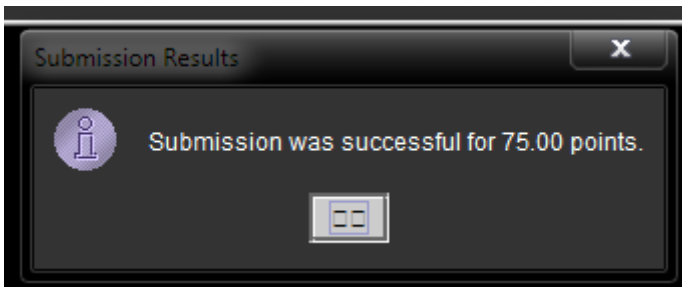
这是什么意思呢？**你能得到多少分只和你交题的快慢有关，但是如果你的程序是错误的（不能通过全部的测试数据），你一分不得，只能对了才能得到这个分数。**所以正确性是最重要的。不要为了抢一点分数而一分不得。



交题之后分数仍然再减。如果你交题之后发现程序有误，想重新提交（需要重新编译），会在这个分数的基础上再减去 10%的罚时。



Resubmit 之后只剩下了 83 分，再 resubmit 一次……



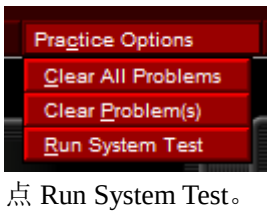
75 分。这就是 250 的保底分。你再怎么重交都不会比这个底了。

接下来你可以开始做 500,1000。

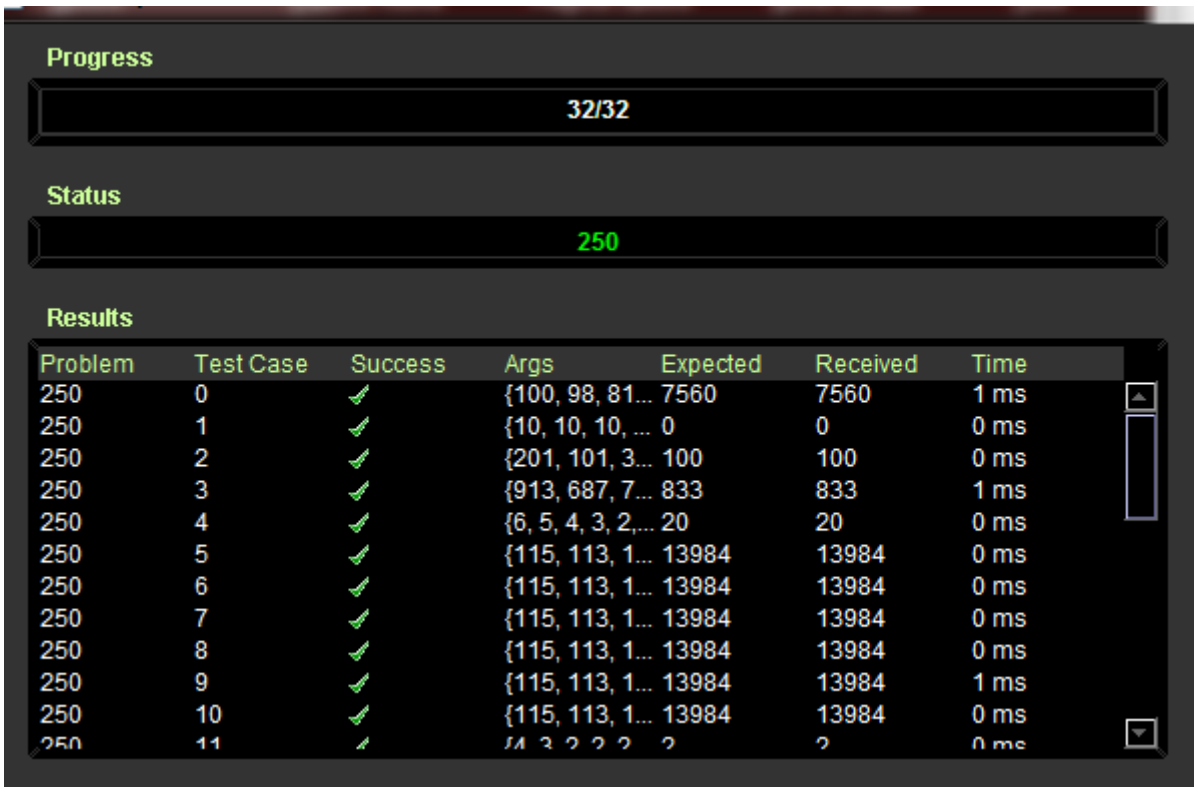
值得注意的是，只有当你打开过某题之后，那题才会才是计时减分，所以做完一题之后没有必要急着开下一题，可以休息一会儿。（练习赛无所谓，但是正式比赛时总时间只有 75min）

好了，我就不演示怎么做 500,1000 了，道理都一样。

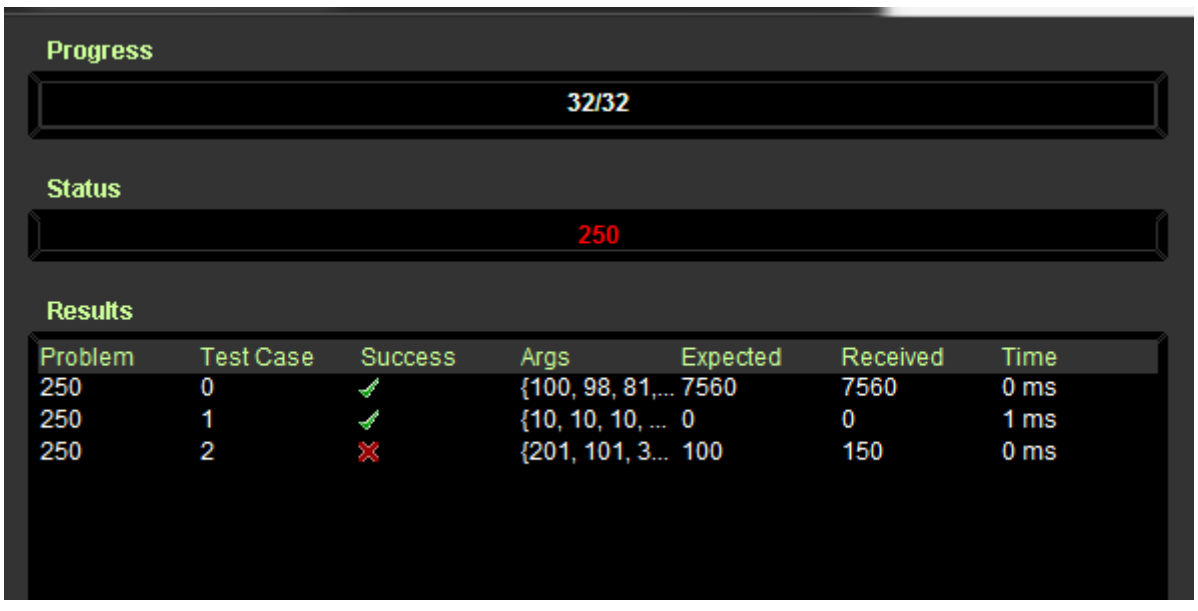
做完之后怎么知道自己的对错呢？



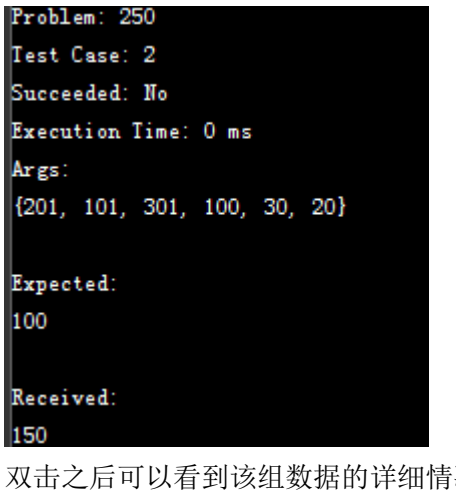
点 Run System Test。



这就是说你 AC 了（绿色）。

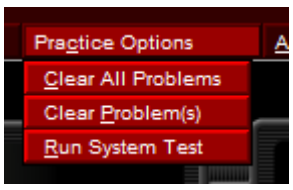


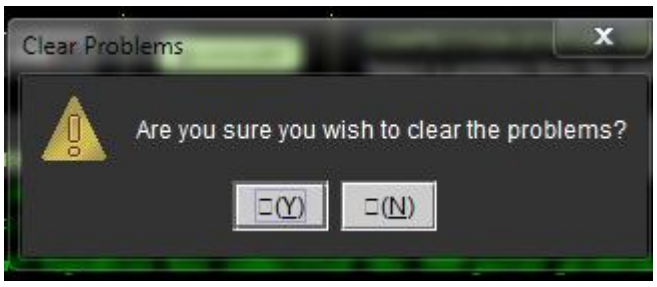
这就是说你挂了。（可以看到第二组就挂了）



双击之后可以看到该组数据的详细情况。

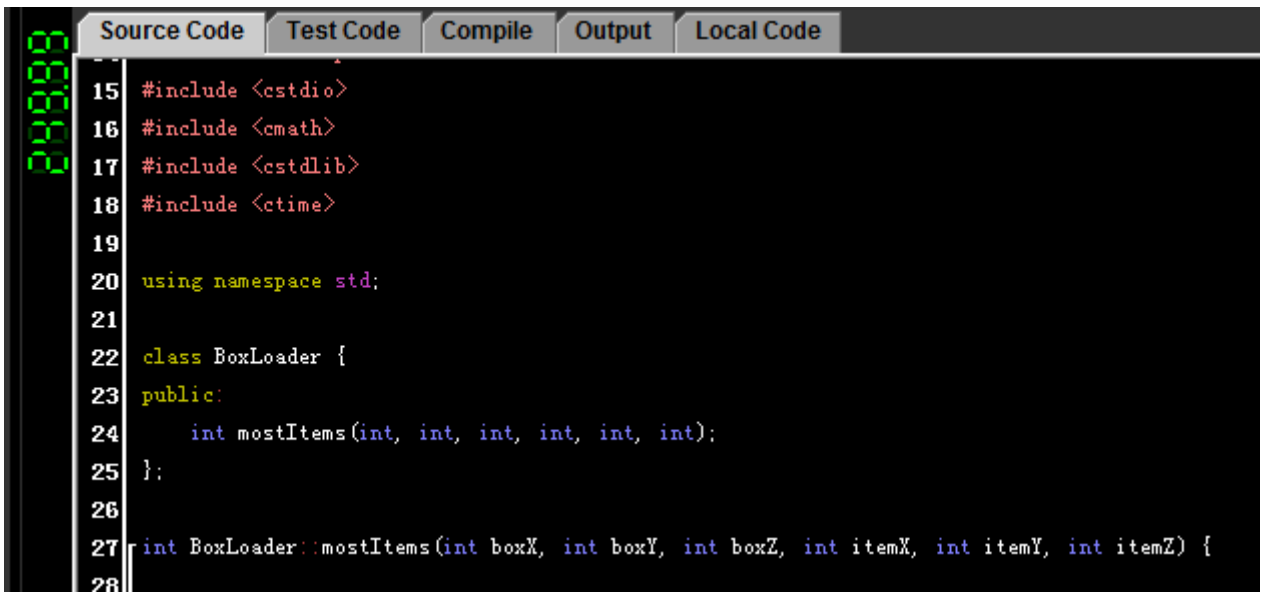
如果你想重新再做这场比赛或者某道题（重新计时）：前两个选项即是。





Yeah！

清除之后编辑器的计时器并不会复原，因为他是另外一套计时体系。如果要一切重来，clear 之后推出 arena 再进去：



复原了。

下次讲比赛：