

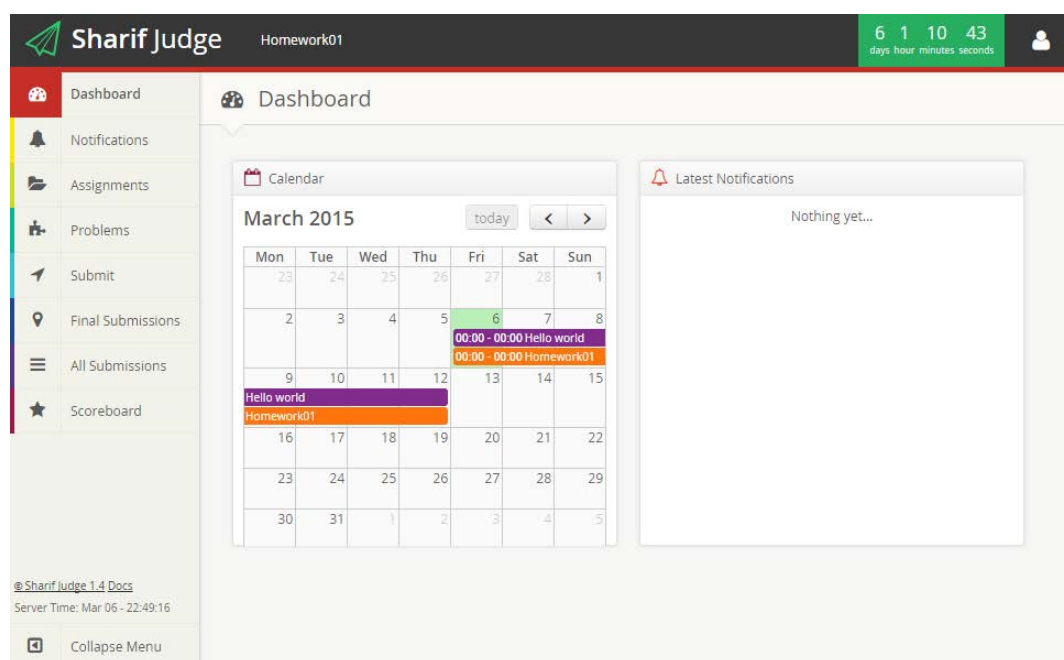
# 数据结构与算法分析

## 作业 01

本课程作业中需要编程实现的部分将使用在线评测（Online Judge）系统进行提交和自动评阅，下文将会介绍在线评测系统的使用，推荐的开发环境，本次作业内容，以及作业提交的注意事项。

## 关于在线评测系统


本课程的在线评测系统位于 <https://oj.iopti.cc/>



## 注册

---

第一次使用时需要注册账号，<https://oj.iopti.cc/register>

The image shows a registration form for 'Sharif Judge'. At the top left is a paper airplane icon, and to its right is the text 'Sharif Judge'. Below this is a form with several input fields: 'Registration Code' with the value '30160192', 'Username (Your Student ID)' with the placeholder '你的学号', 'Real Name' with the placeholder '你的姓名', 'Email' with the placeholder '你的邮箱', 'Password', and 'Password, Again'. At the bottom left is a link 'Login' and at the bottom right is a blue button labeled 'Register'.

Registration Code 为课程编号 30160192

Username 请填写自己的学号，填写错误会影响作业成绩统计

Real Name 请填写自己的真实姓名

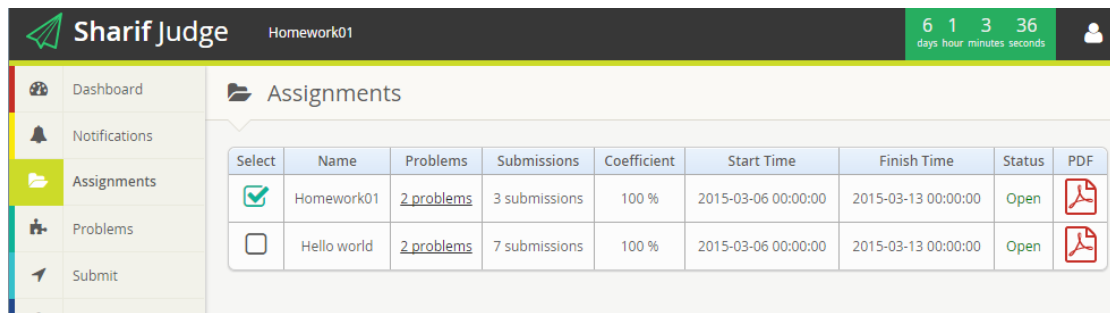
Email 请填写自己的常用邮箱，主要用于密码找回

几点提示：

- 邮箱和姓名如果填写有误，可以在登录之后前往 <https://oj.iopti.cc/profile> 修改
- 用户名如果填写有误，请在姓名一栏中标明有误（方便我们清理无效账号），并将邮箱修改成其他邮箱（因为同一邮箱只能注册一个账号），然后用正确的学号作为用户名重新注册
- 之前选修过本课程的同学可能已经注册了本系统，请使用原有账号登录
- 预计会在本次作业结束后关闭注册，请及时注册账号，以免影响之后的作业提交

## 使用

点击左侧菜单中的 **Assignments** 或者顶部的作业名称可以选择作业，选择作业后在右上角可以看到本次作业代码部分的剩余时间。其中 **Homework01** 为本次作业，**Hello world** 中包含两道基础题（选做），这两道测试题的说明中附有样例代码，大家可以用来热身，以及熟悉评测系统。



选择了 **Assignment** 之后点击左侧的 **Problems** 可以查看作业中的题目



右侧是题目导航以及提交窗口，在这里选择自己编写好的源代码，点击 **Submit** 便可以提交。提交之后显示 **Pending** 表示系统正在准备测试

Sharif Judge Hello world 6 0 58 56 days hours minutes seconds

Dashboard Notifications Assignments Problems Submit Final Submissions All Submissions

All Submissions [Excel](#)

All Submissions of Hello world

⚠ You cannot change your final submissions after assignment finishes.

Final	Problem	Submit Time	Score			Language	Status	Code
			Score	Delay %	Final Score			
○	1	2015-03-06 23:00:53	0	No Delay 100%	0	C	PENDING	<a href="#">Code</a>
✓	2	2015-03-06 21:54:21	100	No Delay 100%	100	C	100	<a href="#">Code</a>

等待一段时间后刷新页面，可以查看评测结果

Sharif Judge Hello world 6 0 57 55 days hours minutes seconds

Dashboard Notifications Assignments Problems Submit Final Submissions All Submissions

All Submissions [Excel](#)

All Submissions of Hello world

⚠ You cannot change your final submissions after assignment finishes.

Final	Problem	Submit Time	Score			Language	Status	Code
			Score	Delay %	Final Score			
✓	1	2015-03-06 23:00:53	100	No Delay 100%	100	C	100	<a href="#">Code</a>
✓	2	2015-03-06 21:54:21	100	No Delay 100%	100	C	100	<a href="#">Code</a>

如果一道题提交了多次，请注意在 **Final** 一列可以点击选择最终版本，也就是实际用来计分的版本（不一定是最后一次提交），每次作业完成后请确认自己选择了正确的版本

Final	Problem	Submit Time	Score			Language	Status	Code
			Score	Delay %	Final Score			
○	1	2015-03-06 23:03:58	0	No Delay 100%	0	C	Compilation Error	<a href="#">Code</a>
○	1	2015-03-06 23:03:20	0	No Delay 100%	0	C	0	<a href="#">Code</a>
✓	1	2015-03-06 23:00:53	100	No Delay 100%	100	C	100	<a href="#">Code</a>

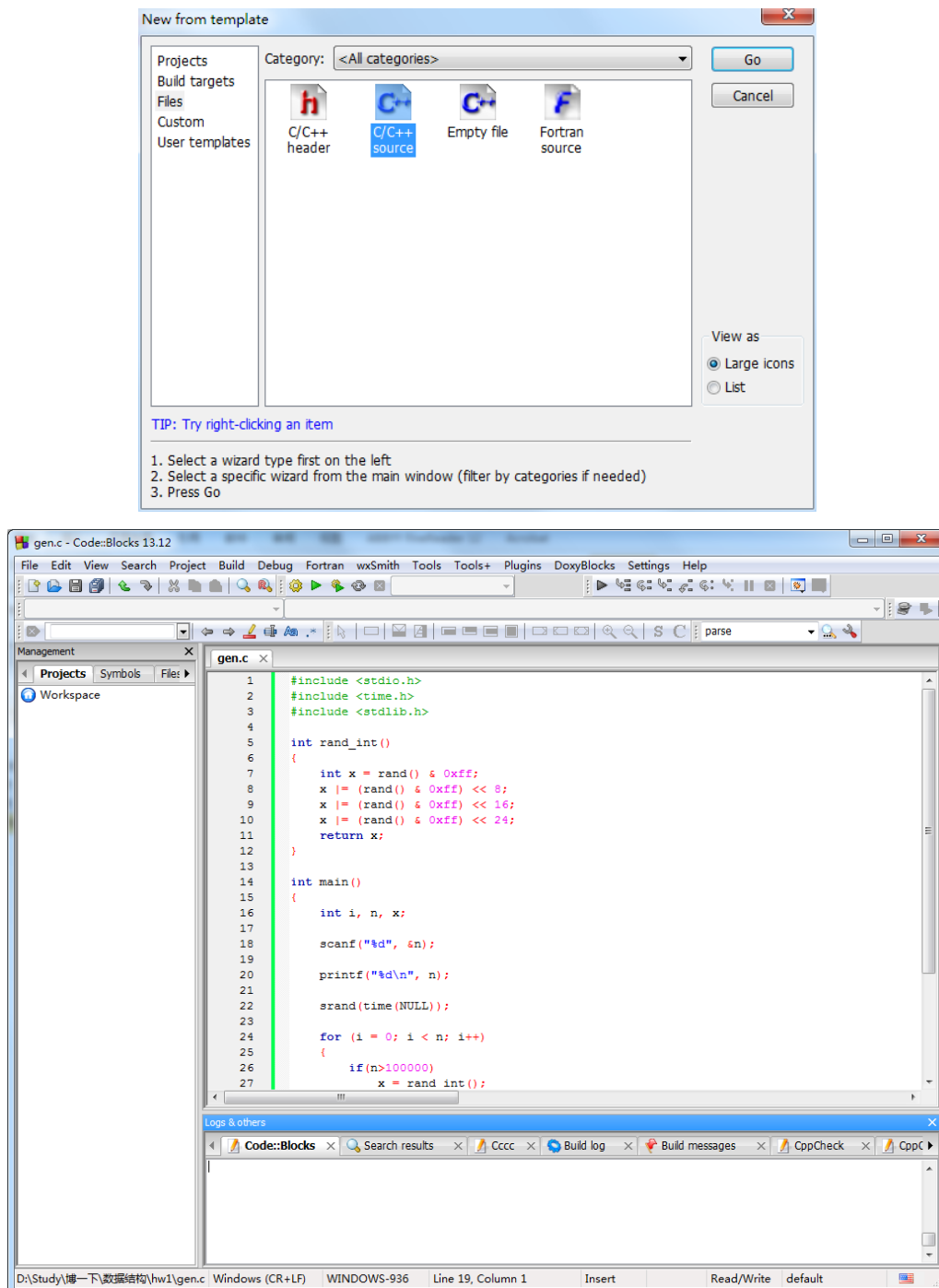
## 推荐开发环境

服务器上使用的是 Linux 操作系统，使用 gcc 对你的程序进行编译。这里推荐大家在自己的机器上使用的是一个轻量的 C 语言集成开发环境——Code::Blocks，官方网站为 <http://www.codeblocks.org/>。

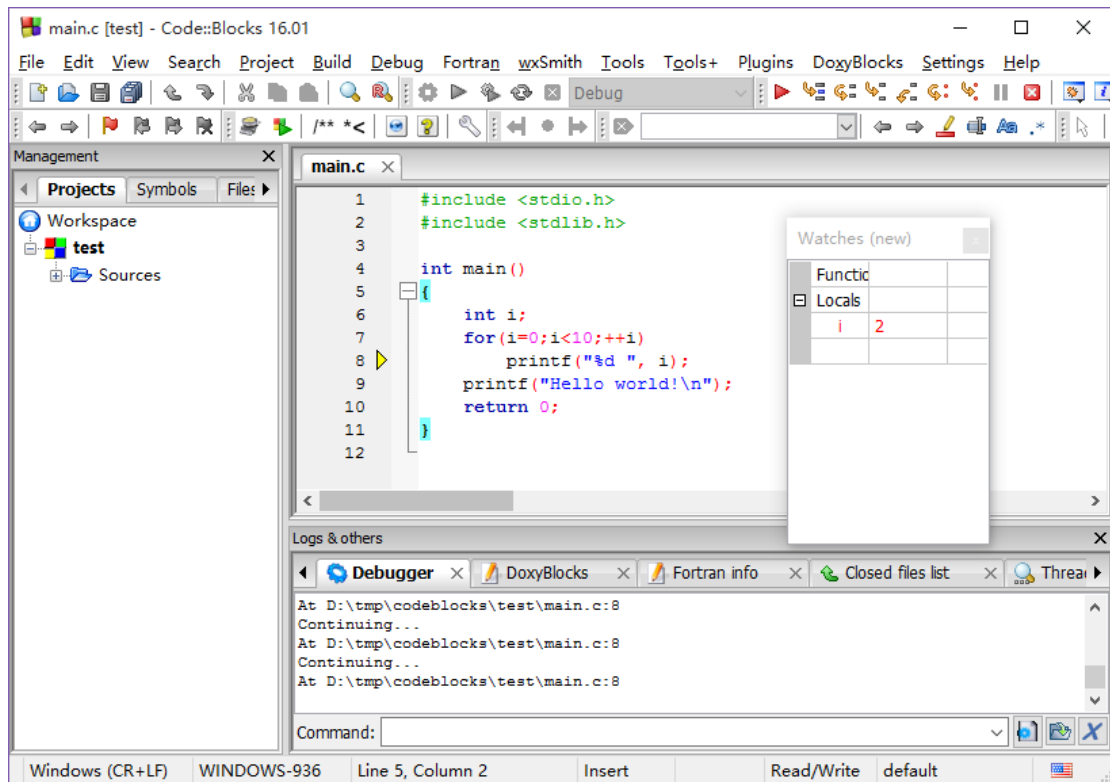
下载页面为 <http://www.codeblocks.org/downloads/26>，Windows 用户请选择 codeblocks-???mingw-setup.exe(???部分是版本号)，这是带有 mingw 的版本(mingw

中含有 gcc 编译器)。目前官网的下载链接正在维护,大家可以从这个地址下载:  
<http://ftp.iethu.info/incoming/3.%E8%BD%AF%E4%BB%B6%E4%B8%8A%E4%BC%A0/Develop/codeblocks-16.01mingw-setup.exe>

软件的使用比较简单,作为一个轻量的开发环境,Code::Blocks 可以新建独立的 C 程序文件,之后进行编辑、编译、测试即可。



注意,如果想要使用调试功能(debugger),比如断点(breakpoint)、单步跟踪(step)、监视变量(watch),你需要创建一个工程,另外你的工程所在的路径不能包含中文,Code::Blocks 的调试器可能无法完善处理中文路径中的项目。



当然，如果你有其他称手的开发环境，也可以使用，只要最终的程序能够通过在线评测系统测试即可。

## 作业内容

本次作业有三道题：

1. 编写选择排序（Selection Sort），详见评测系统
2. 编写归并排序（Merge Sort），详见评测系统
3. 自己生成不同规模的数据，统计并对比上述两种算法的运行时间与数据量的关系

## 关于运行时间的测量

---

关于如何测量一段代码的运行时间，下面的代码展示了最基本的写法：

```
#include <time.h> // 时间相关的头文件

int main ()
{
    clock_t begin = clock(); // 记录开始时间

    // 这里是你想要测量运行时间的代码

    clock_t end = clock(); // 记录结束时间

    double time_spent = (double)(end - begin) / CLOCKS_PER_SEC; // 换算

    return 0;
}
```

在实际使用中，有一些需要注意的地方，例如需要排除数据输入输出等无关时间。如果你想要测量的代码运行时间较短，无法准确测量，除了增大数据规模以外，也可以考虑重复运行测量累计时间再平均。下面是一个更复杂的例子，展示了以下几个要点：

- 排除无关代码的运行时间，例如输入输出、生成数据
- 如果待测量代码运行时间较短不易测量，可以重复运行再平均
- 如果待测量代码需要输入较多数据，而且运行过程中会修改这些数据，那么你可以提前准备好多份数据来避免在重复测量的过程中引入初始化数据的时间

```
#include <stdio.h>
#include <time.h> // 时间相关的头文件

// 我们希望测量 foo 函数的运行时间，注意该函数会对输入的数组进行修改
void foo(int n, int data[])
{
    int i, j, t;
    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - i - 1; j++)
            if (data[j] > data[j + 1]) {
                t = data[j];
                data[j] = data[j + 1];
                data[j + 1] = t;
            }
}

int main()
```

```

{
    int repeat, repeat_times = 1000; // 用于控制重复测量的变量
    int n, i;

    // 注意避免将输入输出等无关时间包括在内
    scanf("%d", &n);
    int* data = malloc(sizeof(int) * n);
    for (i = 0; i < n; ++i)
        scanf("%d", data + i);

    // 提前准备好重复测量中所需要的数据
    int* prepared_data = malloc(sizeof(int) * n * repeat_times);
    for (repeat = 0; repeat < repeat_times; ++repeat)
        memcpy(prepared_data + n * repeat, data, sizeof(int) * n);

    clock_t begin = clock(); // 记录开始时间

    // 重复运行取均值
    for (repeat = 0; repeat < repeat_times; ++repeat) {
        // 这里是希望测量运行时间的代码
        foo(n, prepared_data + n * repeat);
    }

    clock_t end = clock(); // 记录结束时间

    double time_spent = (double)(end - begin) / CLOCKS_PER_SEC /
repeat_times;

    printf("It took %f seconds to run foo(%d)\n", time_spent, n);

    return 0;
}

```

## 注意事项

---

1. 作业中的 1、2 两题需要在评测系统中提交，成绩以评测系统成绩为准
2. 第 3 题可以写成一份简单的文档，通过图表等方式进行对比，并分析总结
3. 作业完成后请将第 3 题的文档和 1、2 两题的最终代码打包提交到网络学堂，用于存档和给出成绩
4. 评测系统使用的测试数据在最终评阅作业时可能会进行调整，提交时的评测结果仅仅是方便大家检查代码，所以请确保程序正确，以免在换用了更完善的测试数据时损失分数



5. 本课程有多位助教，为确保你的邮件得到及时处理，日常答疑邮件请发送至 [DSATHU@126.com](mailto:DSATHU@126.com)，并抄送给姜海老师（CC: [haijiang@tsinghua.edu.cn](mailto:haijiang@tsinghua.edu.cn)）