

前端技术栈

基于Node.js的React框架开发，使用Ant Design及其衍生产品的UI界面库，兼顾图形美术设计与用户友好的界面布局。

后端技术栈

- 后端使用**Python**的**Django**框架，使用**Mysql**作为主数据库，**Redis**作为缓存数据库，**RabbitMQ**管理数据库与测评机。由于使用**ORM**框架，可以无缝支持**PostgreSQL,SQLite,Oracle**等常见数据库。
- 测评端使用**Docker**，**Python**构建。
- 查重使用**Sim**与**MOSS**（目前不支持在线查重，但可通过脚本批量查重）

界面与功能展示

首页

- 全局公告板

题目列表页

- 基本信息
 - 游客可见
 - 题目编号
 - 题目名称，点击可进入题目详情页面
 - 难度等级
 - 通过率
 - 登录用户额外可见
 - 本题状态（未尝试\尝试未通过\通过）
 - 是否位于“喜爱”列表
 - 是否位于“计划中”列表
- 详细信息
 - 普通用户可见
 - 来源标签
 - 算法标签
 - 用户操作菜单（设置喜爱\设置计划）
 - 管理员额外可见
 - 题目可见性（管理员可看到不可见题目）
 - 管理菜单（调整题目可见性、删除题目等）
- 右上角额外菜单
 - 登录用户可见
 - 全局筛选器（全部题目\未通过\喜爱\计划中）
 - 管理员可见
 - 题目操作菜单
 - 添加题目功能（支持匿名和不同类型题目）
- 左侧题库列表

- 可嵌套多题库支持

题目详情页

- 描述分页
 - 题目编号、名称
 - 题目描述 (Markdown解析)
 - 右侧信息边栏
 - 所属题库
 - 题目类型
 - 提供者
 - 时间限制与空间限制
 - 难度等级
 - 来源标签
 - 算法标签
 - 右侧细节边栏 (登录用户可见)
 - 个人历史最高分
 - 个人本题尝试次数
 - 本题目全局提交数\通过数
 - 本题目全局通过率
- 提交分页 (登录用户可见)
 - 编程语言选择 (支持多种语言)
 - 代码编辑 (含对应语言高亮支持, 限JetBrains风格) 与提交
 - 代码文件上传
 - 个人本题目历史提交时间线 (含最近7次提交的得分、结果、时间以及测评详情页链接)
- 管理分页 (管理员额外可见)
 - 题目描述编辑器 (Markdown编辑器)
 - 可预览的Markdown编辑, 支持单页预览和编辑同时预览
 - 修改后可选保存或不保存重置
 - 右侧信息边栏
 - 题目名称、所属题库编辑
 - 时间限制与空间限制编辑
 - 可见性与题目难度评级调整
 - 动态增删来源与算法标签
 - 右侧数据边栏
 - 数据文件列表与大小显示
 - 支持上传单个文件、上传压缩包并自动解析数据文件
 - 支持选定文件的删除
 - 右侧重测边栏
 - 支持按选定评测状态筛选, 并提供便捷的集合化选项
 - 支持按时间区间筛选
 - 支持按用户筛选
 - 支持查询指定条件下的评测记录数量
 - 支持一件重测指定条件下的所有评测记录

测评列表页

- 基本信息分页
 - 游客可见

- 评测ID (超链接, 点击可进入详情界面)
- 评测提交者
- 评测时间
- 题目编号与名称 (超链接, 点击可进入题目页面)
- 评测结果 (支持多种评测结果; 超链接, 点击可进入详情界面)
- 分数 (支持自定义评分细则; 超链接, 点击可进入详情界面)
- 提交所用编程语言
- 部分测评信息 (最大所耗时间、最大内存占用、代码长度)
- 管理员可见
 - 评测时间戳
 - 单条重测功能
 - 单条记录删除功能
- 个人提交分页 (登录用户可见)
 - 本用户的提交记录列表

测评详情页

- 评测结果分页
 - 编译结果与信息
 - 各测试点单点测评结果、时间与内存占用
 - 展开可见测评信息
 - 右侧信息边栏
 - 提交时间
 - 测评题目 (含超链接, 可点击跳转到题目详情页)
 - 提交用户名
 - 评测结果
 - 得分
 - 右侧**本人**历史提交边栏 (同上)
- 源代码分页 (仅提交者本人和管理员可见)
 - 提交源代码
 - 右侧信息边栏
 - 提交编程语言
 - 部分测评信息 (最大所耗时间、最大内存占用、代码长度)
 - 右侧**本人**历史提交边栏 (同上)

课程列表页

- 课程列表分页
 - 右侧学期筛选器
 - 课程卡片 (包含课程标识、课程名称、学期、编号、当前成员数量、简要描述等信息, 点击可进入课程详情页)
- 个人课程分页 (登录用户可见)
 - 本人加入的课程展示

课程详情页

- 课程信息页头
 - 教师姓名、QQ\微信群组号码、助教\学生助理列表
 - 简要课程描述
 - 当前成员数

- 当前管理员数
 - 申请加入\旁听课程按钮（如用户已在课程内，则为提问\退出课程按钮，如用户为课程内部管理员，则为提问\管理按钮）
- 公告板分页
 - 课程公告，课程内部管理员可编辑
- 作业分页（仅课程成员可见）
 - 单个作业标签
 - 作业状态（隐藏（仅课程管理员可见）\未开始\进行中\打折中\已结束）
 - 作业描述（支持Markdown解析）
 - 作业要求（起止日期、评分细则等，支持Markdown解析）
 - 题目列表（含每题得分与总评得分百分比）
 - 修改按钮（课程管理员可见）
 - 右侧成绩边栏
 - 展示本人在课程内的成绩
- 内部题目分页（TODO）
- 管理分页（TODO）

登录页

- 忘记密码与邮箱验证找回功能
- 自主验证码生成

注册页

- 邮箱验证功能

测评部分

JCoder的测评端主要由三部分组成：**Outter_Judger**、**Inner_Judger**与**Config**。其中**Outter_Judger**负责与后端与数据库交互。由于多线程的使用，**Outter_Judger**也可以作为测评集群的管理端，负责统计与任务分发。

JCoder测评端从设计之初便有着**全功能**，**分布式**的设计思路。后端可以与多个不同的测评机建立连接构成分布式测评系统。

Inner_Judger受**Outter_Judger**的管控，运行在我们设计好的**Docker**镜像中。对于**Inner_Judger**的设计，**JCoder**参考了多个开源OJ，并结合大型商业OJ：**luogu**的第四代测评机构建方法，设计出了兼具安全稳定与高效率的测评系统。在**Docker**内部，程序的运行与编译均由**Inner_Judger**全程托管监控。同时采用**Linux**权限控制指令限制了程序的内存使用，IO访问，CPU时间，REAL时间，输出大小，权限等级，网络访问.....等。不仅如此，用户提交的代码在编译运行的全部过程中UID与PID均被锁定，进一步减少程序越狱的可能。此外，**JCoder**还通过**change_root,cgroup**等方式严格限制了代码权限。

为了达到教学目的，**JCoder**相较其他开源OJ系统额外限制了一些奇技淫巧。例如多线程优化，编译优化等。

除此之外，通过自定义配置文件，**JCoder**可以支持多种对比方式与评分标准。例如，您可以自定义每一个测试点的得分与限制，同时可以为不同语言提供不同的时间限制。也可以很方便地支持**Special Judge**。

对于没有特殊需求的题目，管理员可以通过题目管理页面上传测试数据压缩文件。**JCoder**将自动识别其中的输入输出文件并生成对应的配置文件。对于有特殊需求的题目，您需要在上传的**zip**文件中包含名为**config.json**的配置文件，用来指定测试数据与限制。对于输入输出文件，**JCoder**不限制文件名，但需要保证同一组输入输出文件的文件名相同。此外，输入数据的后缀名必须为**.in**，输出数据的后缀名为**.out**，对于不匹配的数据不会被解压。


```
1,
1,
1
], 'memory_limit': [
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1
]
}
}
```

对于**Rejudge**功能的使用，您可参照以下说明。

如果您是管理员，您可以在**submission**页面和**题目管理**页面进行重测。

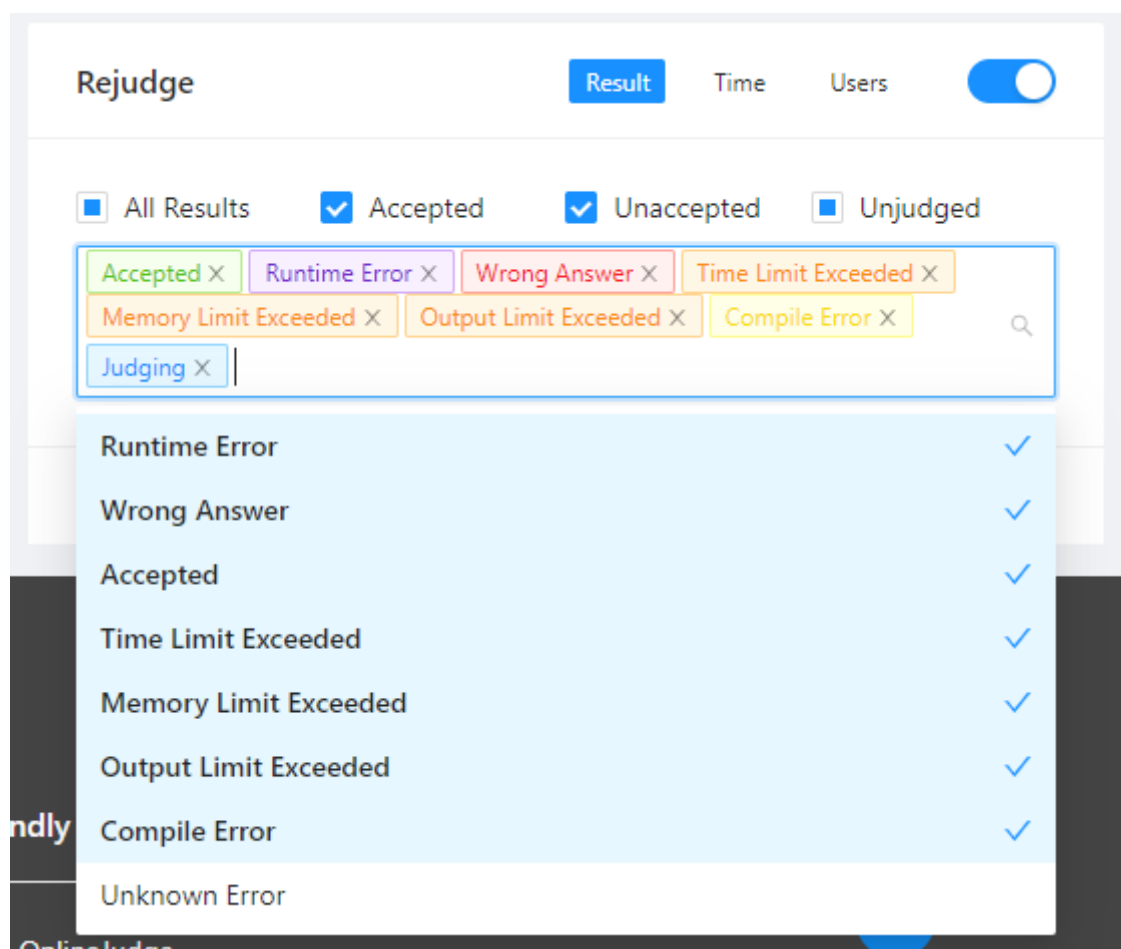
submission页面重测如下图所示：

Timestamp ID	ID	Author	Submitted Time	Problem ID & Name	Result	Score	Language	Property	Actions
1611071267042	2127	11912823	2021-01-19 23:47:47	L101: A*B Problem	Time Limit Exceeded	0	Java	⌚ 3000 ms 📄 8.973 MB 🗑	<div>Are you sure rejudge this submission? <input type="button" value="No"/> <input type="button" value="Yes"/></div>
1611071266319	2126	11912823	2021-01-19 23:47:46	L101: A*B Problem	Time Limit Exceeded	0	Java	⌚ 3000 ms 📄 8.891 MB 🗑 0.295 KB	<input type="button" value="↺"/> <input type="button" value="🗑"/>

使用此方法将重新测试该条记录并覆盖原先的结果。此方法不影响学生尝试次数，但会影响**学生成绩**，**题目AC率**等信息。

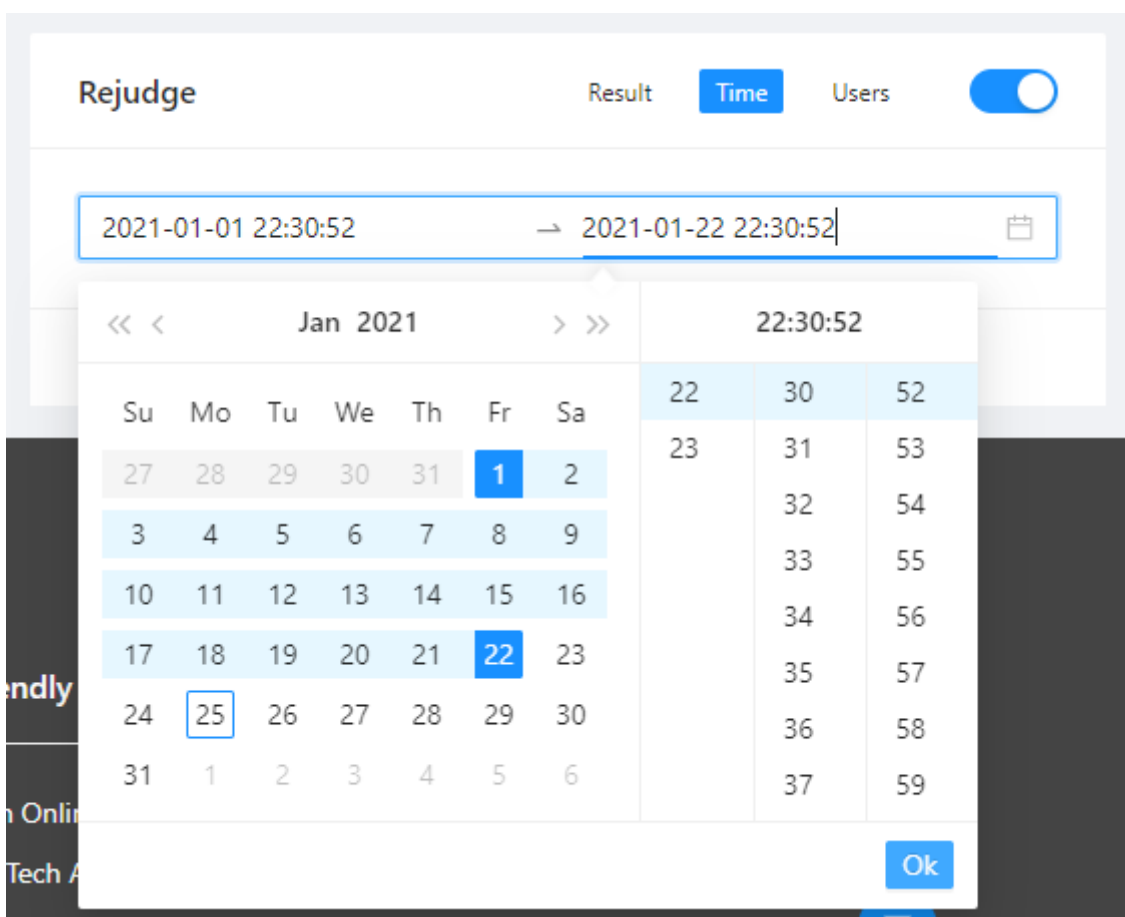
题目管理页面重测由三部分组成，可以选择特定结果，特定时间，特定用户的提交进行重测。重测结果同样不影响学生尝试次数，但会影响**学生成绩**，**题目AC率**等信息。

结果筛选器如下图所示：



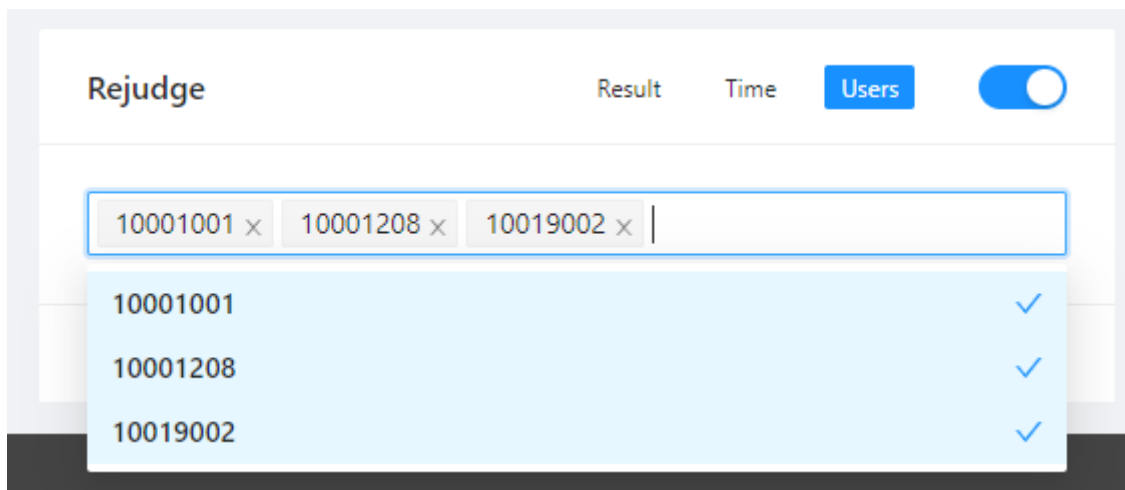
管理员可以在下拉列表选中需要重测的状态，进行测试。也可以使用多选框快速选中对应状态。注意：多选框和下拉列表属于**联动关系**，改变相互影响。

时间筛选器如下图所示：



管理员可以选择特定时间段进行重测。时间精确到秒。

用户筛选器如下图所示：

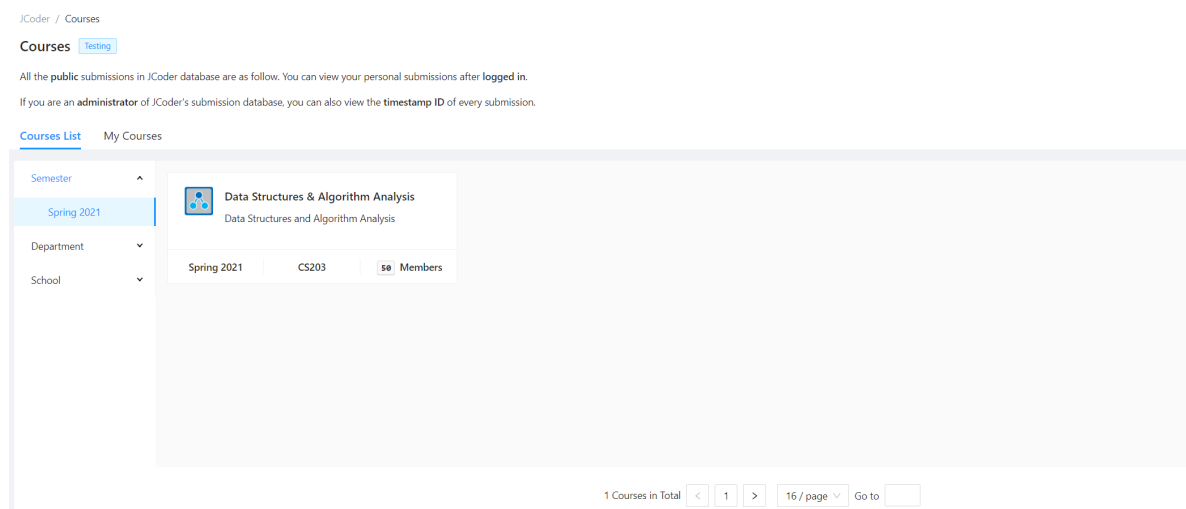


管理员可以通过输入用户名筛选特定用户进行重测。此控件支持**Excel**表格粘贴，将会自动拆分去重。

需要注意的是，虽然以上三种筛选器并没有明确的优先级之分，但是测评结果筛选器为必填，时间与用户筛选器为选填。默认为所有时间与所有用户。对于不存在的用户，虽然能填入用户筛选器，但会在后端被忽略。

课程部分

JCoder允许管理员创建课程以适应教学用途。现阶段，您可以在课程中**发布通知**，**布置作业**。在后期的开发过程中**JCoder**会参考**Sakai**与**Blackboard**系统，引入完整的文件服务与课程管理平台。



课程内部权限控制与外部稍有不同。权限暂时分为两类：**课程管理员**，**课程学生**。**课程管理员**在课程内部具有类似**全站管理员**的权限，可以**添加题目**，**创建**，**管理作业**，**查看统计结果**等。考虑到安全性与教学需要，目前**JCoder**只允许**全站管理员**查看代码。

JCoder课程内部页面如下图所示：

JCodb2021-β

Instructor: AnonymousSocial Media Group: 489324857TA\SAs: 1191282311912824

Invrise

Course Descriptions

test_info

AnnouncementsHomeworksProblemsDiscussionMembers

Enjoy DSAA

Top

Posted by satan2021-01-26 13:17:14

test01	test02	test03
test	test	test
test	test	test
test	test	test

In-Course Personal Details

No Data

第一次作业成绩通知

Posted by Invrise2021-01-26 20:07:59

第一次作业DDL已过，SA已进行了分数的核对以及分数线的划定，现公布成绩，同学们可去往 Course -> Homework -> My Grades 查看本次作业的成绩。
如有任何问题，请及时联系SA，以便核验和修改错误。
祝好。

JCoder

Posted by satan2021-01-26 12:55:21

All the public submissions in JCoder database are as follow. You can view your personal submissions after logged in. If you are an administrator of JCoder's submission database, you can also view the timestamp ID of every submission.

JCoder的课程通知页面如图所示。通知主要分为**标题**，**内容**两部分。其中**内容**部分支持完整的**Markdown**语法。

使用通知功能时需要注意以下几点：

- **课程管理员与全站管理员**均可在课程内部发布，管理公告。

Ask QuestionsManage

📅 Add an Announcement

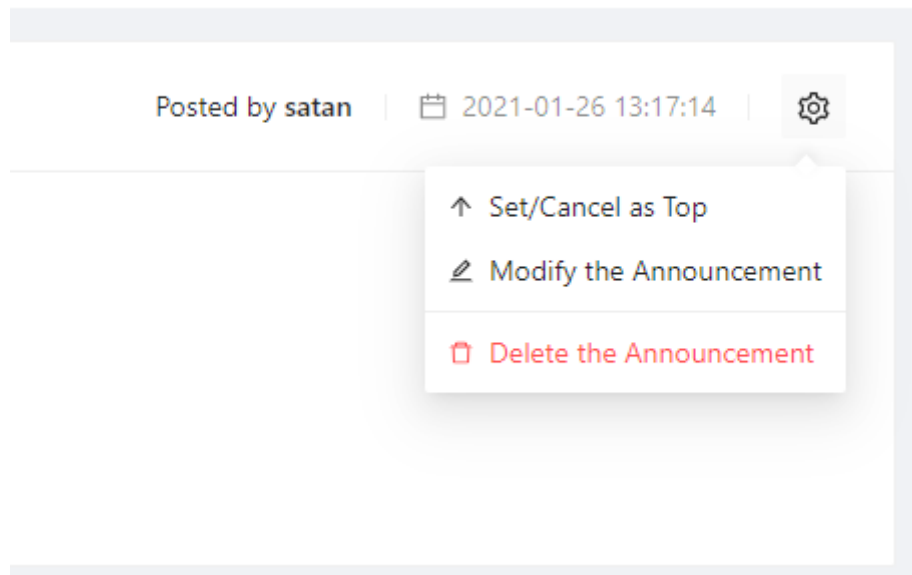
✎ Add a Problem

📝 Add a Homework

🔧 Modify Course Information

🗑 Delete the Course

- 公告发布后，管理员可通过管理按钮随时更改公告的**内容**，**标题**，**是否置顶**等特性，也可随时**删除**公告。



social Media Group: 489324857

绩, 同学们可去往 Course -> Homework

Modify the Announcement

Editor Preview Editor & Preview

Title: Enjoy DSAA

test01	test02	test03
test	test	test
test	test	test
test	test	test

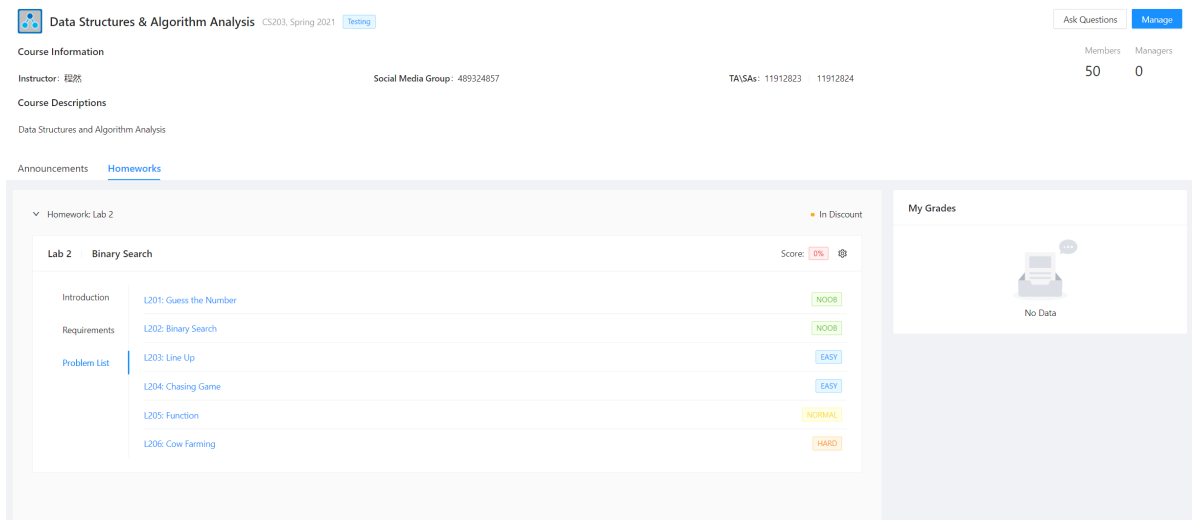
test01	test02	test03
test	test	test
test	test	test
test	test	test

Cancel

Reset

Save

- JCoder的公告按照发布时间倒序排布，当前版本不支持拖拽与自定义顺序。



在**JCoder**的课程详细界面，顶部**信息栏**将显示显示课程相关信息。

下方**Homework**选项卡页面中会显示当前课程的**作业信息**。

JCoder的作业信息由**编号**，**名称**，**介绍**，**要求**，**题目列表**等部分组成。其中**介绍**与**要求**部分同样支持**Markdown**语法。

不同于大部分开源OJ，**JCoder**有四种作业状态：**未开始**，**进行中**，**延期**，**结束**。会在作业的**右上角**标注。

- 当作业处于**未开始**状态时，此次作业**只有管理员可见**，非管理员的**提交**，**强行查看**均被禁止。
- 当作业处于**进行中**状态时，如果此次作业被标记为**可见**，则非管理员用户可以**查看并提交**题目。请注意，**作业可见性的优先级大于!!!作业状态!**
- 当作业处于**延期**状态时，权限与**进行中**一致，但会在成绩统计界面进行标注，提供不同的成绩统计方式。
- 当作业处于**结束**状态时，非管理员用户的提交被禁止，但仍然可以查看题目。

JCoder允许管理员在课程内创建题目，课程内题目**不会**在主题库中显示。对于课程内题目，**JCoder**提供三种不同的创建方式：**复制**，**新建**，**链接**。

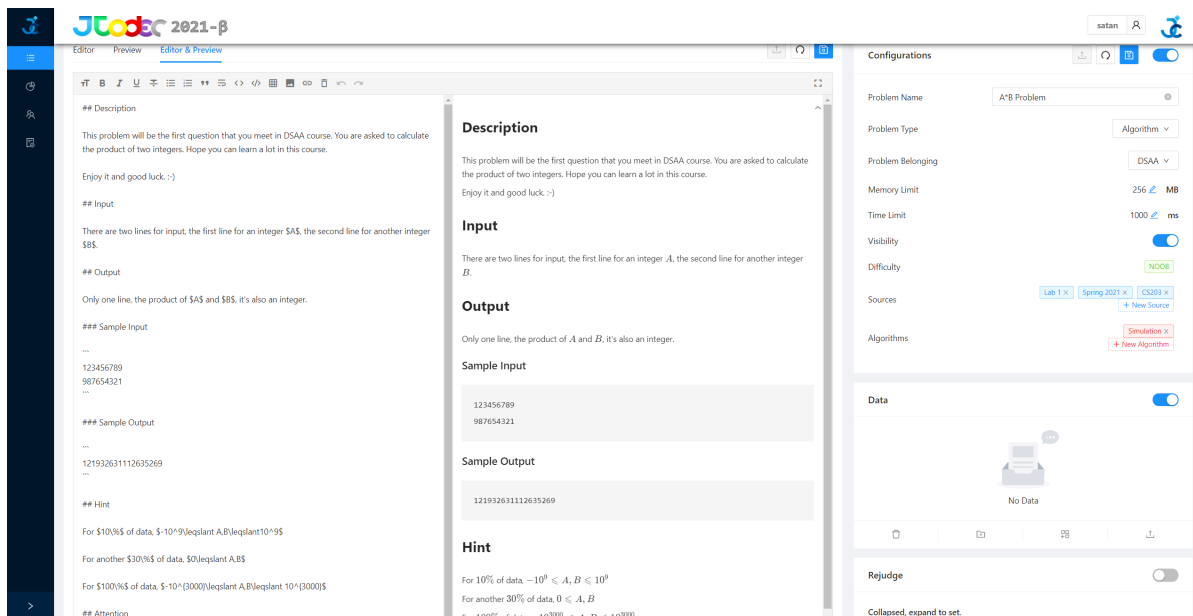
- 如果采用**复制**方式添加题目，管理员可以选择一道**可见的主题库题目**添加进课程。此操作会同时复制题目数据。对于采用此种方式创建的题目，课程内管理员拥有该题的**完全修改权限**。如**更改题目**，**更改数据**等。
- 如果采用**新建**方式添加题目，**JCoder**将会新增一道**空白题目**。与**复制**类似，课程内管理员拥有该题的**完全修改权限**。
- 如果采用**链接**方式添加题目，管理员可以选择一道**可见的主题库题目**添加进课程。此操作不会复制题目数据，但会继承**AC率**，**提交次数**等信息。对于采用此种方式创建的题目，课程内管理员**没有**该题的**修改权限**。该题被从外部删除的同时也将从课程内消失。

TODO:

JCoder会为管理员提供课程与作业的统一信息。在目前的设计思路下，管理员可以通过**提交时间**或**提交次数**分段统计成绩，并支持导出。

题目管理

JCoder提供完善的题目管理功能，管理页面如下：

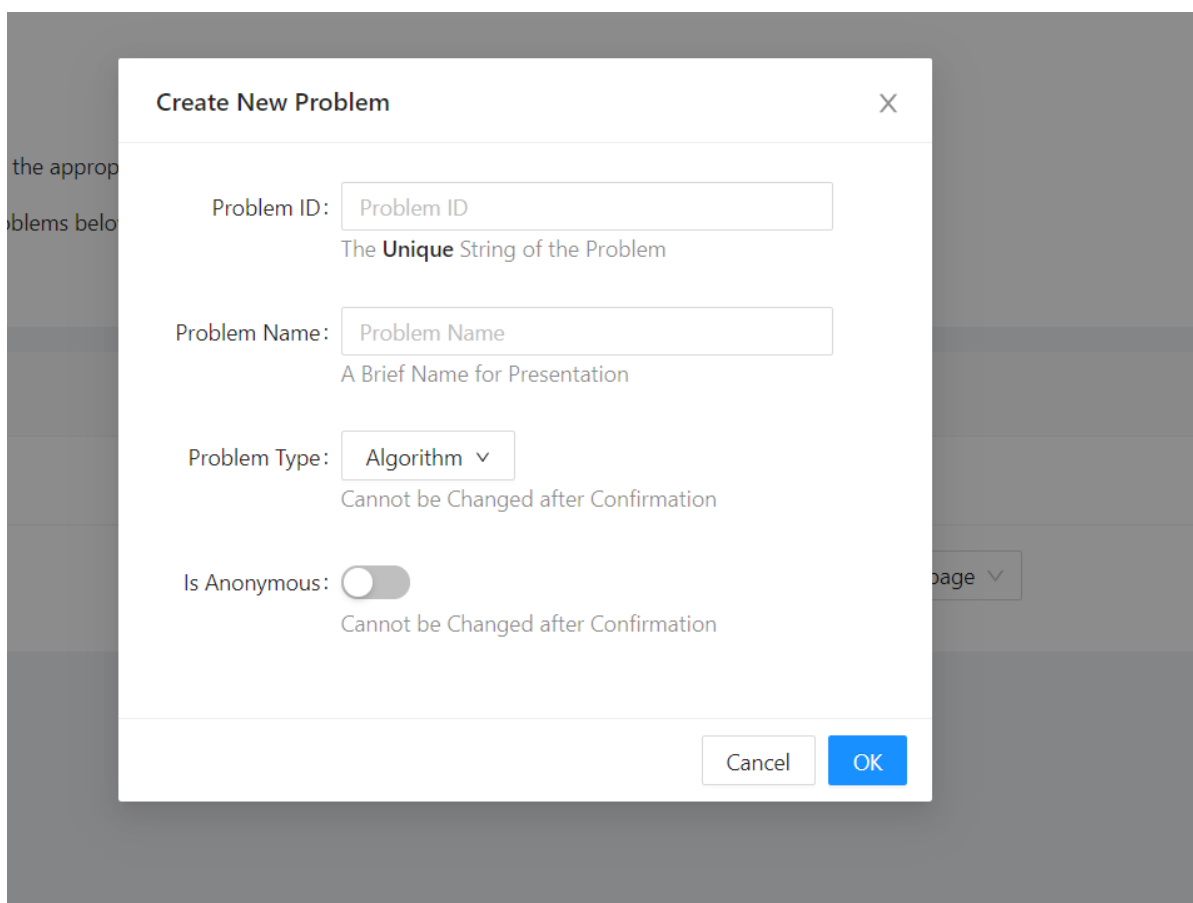


左侧为**题面编辑区**，管理员可以使用**Markdown**语法编辑题面并提供**实时预览**。

右侧为**题目设置**，**数据管理**，**重测分区**。管理员可以随时**更改题目名称**，**时间限制**。也可以**动态添加标签**，**改变题目可见性**，**添加数据**等。

需要注意的是，以上所有操作在保存后会**立即生效**。

对于添加题目，操作方式如下：



管理员需要指定**Problem ID**,**题目名称**,**题目类型**与**是否匿名**。其中除**题目名称**外，其他信息一经确定无法更改。

点击确定后，**JCodeC**将采用默认配置新建一道**不可见**题目供管理员后续修改。

