

2019-2020 学年第二学期

《数据结构》课程设计线上实施方案 V1.0

同学们，大家好，因新型冠状病毒肺炎疫情影响，本学期的《数据结构》课程设计将通过**超星慕课平台**和**校外在线评测系统**线上开展。事实上，即便没有疫情影响，这学期的课程设计也会结合超星和在线评测开展，所以唯一的不同在于编程地点从机房、寝室移到了家里。

考虑到有极少数同学家里没有电脑或网络，所以对于这些同学，可以特事特办，允许开学之后再补交程序。但请注意，开学之后肯定会非常非常忙，所有因延期开学耽误的课会蜂拥而上，恐怕很难有充足的时间和精力高质量完成课程设计，此外，对于开学后提交程序的同学，我们会以最大的力度重点检查，以防止抄袭。所以我还是强烈建议，如果没有极度特殊的困难，还是尽量跟着线上进度完成本课。

本次课设一共有 4 道题目 A、B、C、D，其中 A、B、D 题个人独立完成；C 题以小组形式完成，每组 3 人，在本班内自由组队。4 道题目的程序，每题 100 分，共 400 分。此外，课程设计报告、签到情况、平时表现等共 100 分。总成绩为 500 分。最后根据每班内部总成绩排名评定最终成绩，最终成绩为五档：优、良、中、及、不及。每班具体由 1 位老师负责检查程序和评定成绩。

一、诚信要求

对于要求个人独立完成的题目，同学之间可以交流对题意的理解、解题思路、数据结构与算法的选用与设计、测试用例等。但不可交流源代码。此外，允许在网上查阅、参考相关资料和文献，但不允许抄袭网上代码。

由于本次课设，老师很难像以前那样当面检查同学程序，所以我们会加大力度进行代码的自动查重。**我们仍采用国际上最权威的代码雷同度检测软件**，并结合老师人工评判，不定期对同学的代码进行批量查重。必要时我们也会通过视频答辩形式检查学生是否抄袭。在判定雷同度时 KMP、Dijkstra 等标准算法不计在内。该查重系统已在上学期计算机学院上机实验中**成功应用**。

任何时间被发现抄袭（程序不是自己编写的、与其他同学或网上代码雷同），则该题目不仅成绩取消，并且视情节轻重倒扣分，如记-50、-100、直至-500 分。例如某题记-100 分，则即便其他都得满分，则总成绩最多只能是-100+400=300。如果某题记-400，则意味着本课肯定是 0 分。**抄袭者和被抄袭者双方同论，不做区分，无论是否同班、同专业、同学院。所以请同学们不要坑自己，更不要坑你的朋友。**

查重不定期进行，可能在任何时间，比如你 3 月初完成的程序，我们可能在 6 月份进行查重，届时会对你的成绩进行修正。

二、课程形式

我们基于超星慕课平台发布题目（pdf 格式），同学们按要求编写和提交代码。基于题目性质的不同，本次**学生提交代码**和**老师检查代码**包含两种方式：

（1）**全自动评测**：大家通过线评测系统（jlu.openjudge.cn）提交程序，该系统实现程序的自动评分。请大家在 jlu.openjudge.cn 注册账号，用户名和真实姓名请一定使用真名，并在主页左边“修改设定”中选定自己所在班级，昵称也改为姓名。目前有部分已注册的同学，用户名并不是真名，或者没选定自己班级。届时老师将无法登记这部分同学的成绩，只能暂按 0 分记。

（2）**半自动评测**：有一些题目没有标准答案，答案是多样的，这部分题目暂时没有全自动评测系统，目前是由我编写或提供评测程序。同学们按接口要求，通过超星慕课平台提交代码（一般来说是一个 C++ 函数）。老师收集、整理学生代码，将学生代码（一个或多个 C++ 函数）放入评测程序中，在本机编译、运行评测程序，得到学生分数。

具体题目及评测方式将在课程进行过程中逐一发布。

此外，每题都需提交课程设计报告（解题报告），具体模板将统一发给大家。请认真撰写，不必粘贴代码，只需写解题思路和算法评估分析等，1-2 页即可。课程设计报告将在 100 分的平时成绩中

占据很大比重。

我们将按照课表时间（1-8 周周三下午 13:30 左右）通过超星平台签到。但大家编程时间自由，可以在规定截止期前的任意时间编写和提交程序。

三、在线评测平台 jlu.openjudge.cn 简要使用说明

我们采用了校外可访问的在线评测系统 `openjudge`，该评测系统由北京大学运营。其支持 C/C++、JAVA、PASCAL 语言。如果你使用 C++ 语言，请选择 G++，如果你使用 C 语言，请选择 GCC。将你的源代码粘贴到代码框内，点击提交，等待评测结果。评测结果主要包含如下几种类型：

Accepted: 通过，恭喜。

Wrong Answer: 程序结果错误。原因多在于程序存在 bug 或有些地方没考虑到。

Compile Error: 编译错误。常见原因：（1）程序虽然可以在 Visual Studio 编译通过，但不符合 GCC/G++ 或 C/C++ 标准，例如 Visual Studio 可能会自动 include 一些头文件，GCC/G++ 没有。Visual Studio 推荐使用 `scanf_s`，但该函数不属于 C/C++ 标准，GCC 没有。（2）提交时选错语言类型。（3）不确定编译器是否支持 C++ 11，所以尽量不要使用 `nullptr` 关键字，如想使用空指针可用 `NULL` 或 `0`。等等。请注意 GCC/G++ 与 Visual Studio 有些许差别，一般认为 GCC/G++ 更接近 C/C++ 标准，建议使用 VS 的同学，要创建空工程，提交的代码中不要含有 VS 特有的一些头文件如 `pch.h`、`stdafx.h` 等。

Presentation Error: 格式错误，你的答案和标准答案相比，差一些空格或回车，离 Accepted 已经无限接近了。

Time Limit Exceeded: 超时，程序时间效率不够高，无法在题目规定时间内运行完毕。

Runtime Error: 运行时错误，导致程序异常退出。主要原因包括：（1）数组越界访问，例如题目中写明测试数据最多 100000 个字符，但你只开了 10000 长度的数组，则会导致程序在服务器运行时数组越界。而你在本地并没有用 100000 长的字符串测试，所以在本地没有发现该错误。（2）数组开太大了，超出系统对该题规定的内存限制，例如有的同学开了 `10000*10000` 的数组，但系统对该题内存限制只有 10M。（3）访问了空指针、野指针、已经释放过的内存空间。在编写链表、二叉树程序时，容易出现此错误。（4）递归层数过多，超出系统栈大小限制。（5）另外 GCC 禁止 `void main()`，`main` 函数必须 `return 0`，否则也可能出现该类错误。

此外，对于以前从未使用过 OJ 的同学，提醒大家注意，程序输出必须完全按照题目要求的格式输出，不要输出诸如“请输入 n 和 m....该题的结果为....”之类的辅助性文字。

关于评测系统的其余问题，请见 <http://openjudge.cn/help.html>