

数据结构课程设计

第13周-2018年12月5日

课程设计的预期进度

- 第11周：发布课程设计的题目形式与计分原则
- 第12周：发布课程设计的简易样例
- =>第13周：发布课程设计的提交要求
- 第14周：课程设计中期检查
- 第15周：发布课程设计的最终计分方法
- 第16周：课程设计报告及代码最终提交

- 第17周：预约面谈
- 第18周：预约面谈

目录结构

- 17307130000 (学号文件夹)
 - build.sh
 - main.cpp
 - yourcode/
 - Submit-1AddPointBeforeQueryPolygon.h
 - ...
 - ...
 - Tools (给大家的工具)
 - draw.py 可视化工具
 - score.py 评分工具 (还在写)
 - 其他大家如果还有需要的工具可以提出来, 要求合理的话我帮大家写
 - Data (逐渐为大家提供)
 - ...

PJ 中期检查的任务

- 提交一个可以编译的、按照要求打包的代码，正确性没有要求。
- （注意重点：中间检查是为了保证大家已经有方法提交可以编译、可以运行的代码）
- 可以编译的：在 build.sh 中自行书写编译方法，我的执行方法为
- `$ bash build.sh`
- 如果使用 make 或者 cmake 请将 Makefile 或者是 CMakefile.txt 放置在与 build.sh 同目录下
- 不需要改动 main.cpp 内的代码

PJ 的一些 Tips

- 三个静态任务实质上是等价的
- 优先选择一个比较好的**空间划分方法**和**空间索引数据结构**会让六个任务都比较轻松
- 如果需要使用外部的库函数或者是自己书写的独立数据结构，一个比较好的独立的初始化函数（进行内存的获取以及初始化）可以帮助你更好的组合代码
- 推荐使用 C++ STL，由 STL 提供的可以安全访问的容器可以有效的减少管理内存等多文件工程的实现难度
- 对于较大的结构，如果作为函数的参数，考虑传递引用而不是 copy 副本，copy 副本效率很低，如果需要使用内存 copy 函数（memcpy 等），**选择安全的使用方法可以有效的降低 Debug 难度！**