一个没写论文标题的O奖论文手的自白

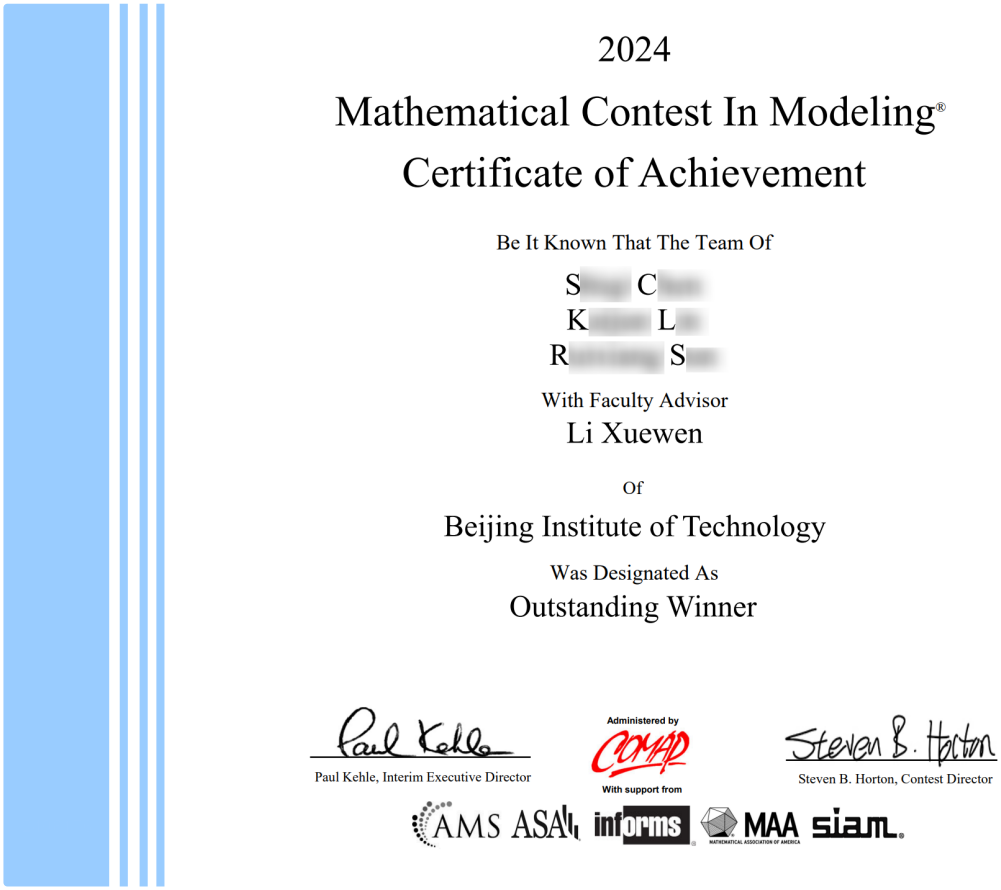
一个没写论文标题的O奖论文手的自白

- 0. 前言
- 00. 资料清单及简介
  - 00.1 百度网盘🔗 💎
  - 00.2 一些不错的经验分享类干货🔗 ✍️
- ∞. 假设
- 1. 数学建模比赛的本质⭐ ⭐ ⭐ 🤔
  - 1.1 不要把数学建模当作本质的原因之二
- 2. 美赛到底考察什么?
  - 2.1 核心
  - 2.2 美术建模? ⭐ 🤔
    - 2.2.1 模型总流程图
    - 2.2.2 核心模型流程图⭐ 😈
    - 2.2.3 但是...🤯
  - 2.3 创新性? ⭐
  - 2.4 可读性
    - 2.4.1 9个操作性极强的小技巧✍️
- 3. 为什么我们拿了O奖? 有哪些优点可复制性极强?
  - 3.0 相信自己，谨慎参考付费思路⭐ 😊
  - 3.1 语言表达⭐
  - 3.2 搜论文⭐
  - 3.3 扎实的建模（概念及应用）、编程基础
  - 3.4 清晰的团队分工
  - 3.5 时间安排及进度把控
    - 最后优化清单
  - 3.6 论文框架⭐ ⭐ ⭐
- 4. 一个优秀的论文手应该做到哪几点?
  - 4.0 语言表达
    - 写提示词的基本规则⭐ ⌨️ 💡
  - 4.1 论文逻辑性、连贯性
  - 4.2 一个合理的写作顺序
  - 4.3 会写摘要...但这并不难！ ⭐ ⭐ ⭐ 🔍
  - 4.4 速读论文
  - 4.5 解决编程手和建模手偶发的沟通困难
  - 4.6 快速、美观绘图
  - 4.7 手要快
- 5. 一位优秀建模手的自白
  - 5.0 前言
  - 5.1 对数模比赛的理解
  - 5.2 成为建模手需要什么
  - 5.3 心得和展望
- 6. 论一个称职的队长
  - 6.0 对队员的绝对尊重和耐心
  - 6.1 果断——把控团队解题进度
  - 6.2 保持冷静
- 7. 如何找队友?
  - 7.0 理想的论文手/建模手/编程手的模样
  - 7.1 陌生人组队⭐ 😈/👶
  - 7.2 熟人组队
- 8. 如何进行备赛最有效?
  - 8.0 对于整个团队：周会 & 模拟赛⭐
  - 8.1 论文手⭐
  - 8.2 建模手
  - 8.3 编程手
- 9. 比赛前，可以为比赛准备什么?
  - 9.1 论文手
  - 9.2 建模手
  - 9.3 编程手
- 10. 比赛过程中我们使用到的工具
  - 10.0 团队论文调研平台
  - 10.1 团队沟通平台
  - 10.2 论文手使用到的

10.3 建模手使用到的
10.4 编程手使用到的
11. 推荐一些公众号
11.1 一个好用的搜索小技巧⭐⭐⭐
12. 美赛带给了我什么？⭐
后记

0. 前言

大家嚎！我是lucky！这是我的第二个经验贴！惯例放图：



2024

Mathematical Contest In Modeling®

Certificate of Achievement

Be It Known That The Team Of

S[redacted] C[redacted]
K[redacted] L[redacted]
R[redacted] S[redacted]

With Faculty Advisor

Li Xuewen

Of

Beijing Institute of Technology

Was Designated As

Outstanding Winner

Paul Kehle, Interim Executive Director

Administered by COMAP

With support from AMS ASA IMS INFORMS MAA SIAM

Steven B. Horton, Contest Director

是的，这是一个**近三万字！！**的经验贴，可以认为是一个围绕如何在美赛中获奖、但同时又试图向朋友们解释数学建模本质的文章。这并不是数学建模教程，并不能帮助小白**从零**入门（但也许可以帮助论文手入门，此外我在章节**00.2**给出了一些**小白友好的美赛介绍贴**，纯小白可以先看那些）。b站上大量存在的入门级内容，并不是本文叙述的重点（当然，本文也没有什么能让真正懂建模的大佬眼前一亮的内容，还望轻喷）。不过我相信，任何水平的数学建模爱好者完整读完这篇分享贴后，都能够有所收获或共鸣。这篇文章旨在以尽可能简练、准确的语言，从我自己这个懂一点模型和编程的**论文手**的角度，为美赛小白提供一份全面的指南。涉及到论文手的部分非常详细，但涉及建模手、编程手的部分若有疏漏请多包涵（我并不是合格的建模手/编程手）。文章将分析数学建模比赛的本质，讨论美赛所考察的核心能力，如解决问题、创新性和可读性。接着，我将分享团队获得O奖的具体经验，包括搜论文、扎实的建模和编程基础、清晰的团队分工、有效的时间管理和严谨的论文框架等。本文还将探讨如何成为优秀的论文手与建模手，并提供找队友和高效备赛的必看建议。在比赛准备和进行过程中，本文也将推荐一些工具和资源，最后本文会推荐一些相关的网页资源供读者参考。(summarized by GPT-4o)

本文涉及的一些漂亮图，和我们团队的o奖论文pdf版本，以及其他相关资料，都在如下00. 资料清单及简介**中，请及时保存。若发现链接失效，请vx私信联系，我的vx号是lkj18811012005，若有问题想要咨询，欢迎加我微信。团队论文的latex源文件链接如下：https://www.overleaf.com/read/fpjcfrjrjnrd#cd1c9b（overleaf在线编辑，我设置了只读。比如大家想要找伪代码的源代码，就在右半屏幕的pdf预览中找到，然后双击pdf内的伪代码，overleaf就会自动定位将latex编辑器定位到写相应伪代码的部分，很智能）。**

00. 资料清单及简介

00.1 百度网盘🔗 💎

链接：https://pan.baidu.com/s/1hbpZgFOIY3JV15fNYql1-A?pwd=wv9v
提取码：wv9v

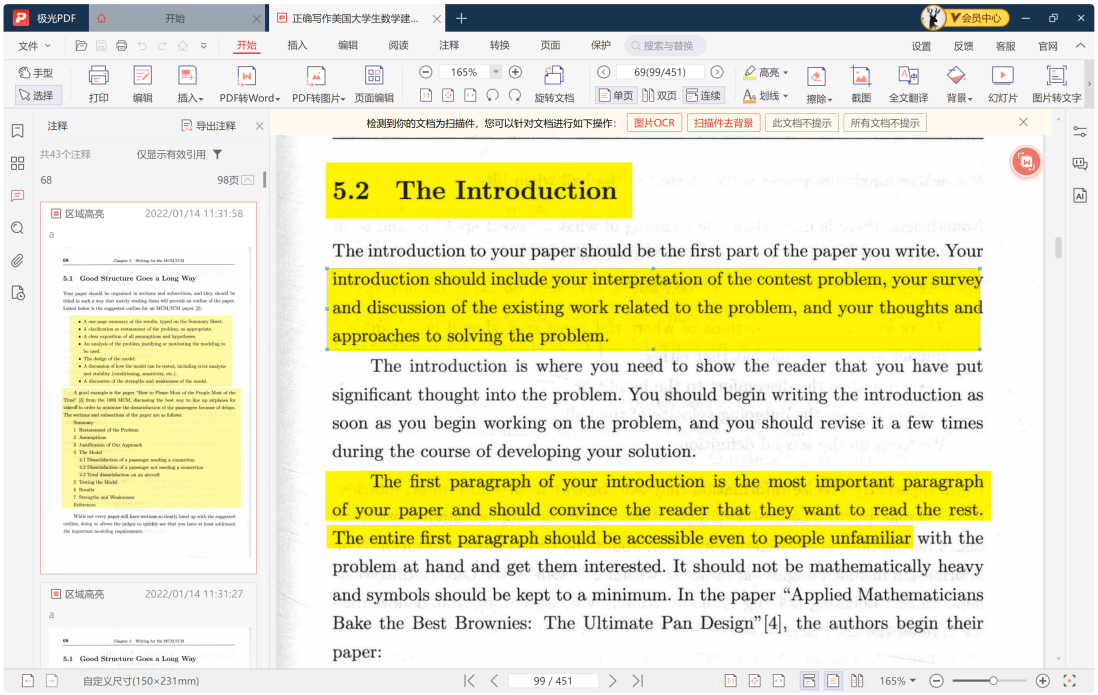
这里介绍**部分**文件：

- 论文手喜欢的.文件夹
  - 各种latex & word写作模板
  - 2224507-C题模板论文.pdf

    也就是我们O奖论文参考的模板文
  - 各种图表.pptx

    我画过的较为通用或者较为好看的一些图片，以及本次比赛的一些结果。**核心模型流程图和模型总流程图都在里面**，甚至还包含了我这次比赛没用到的“数据处理-指标计算-模型建立”流程图，我真的我哭死。
  - 正确写作美国大学生数学建模竞赛论文　第2版＝WRITE RIGHT FOR THE AMERICAN MATHEMATICAL CONTEST IN MODELING_14284249.pdf

    小白论文手必看。如果一个论文手不看这个就去打比赛，那么这个论文手必定是相对失败的（暴论，不至于啦）。我对我认为比较关键的内容都进行过注释，大家如果没有时间，可以直接去看我高亮的部分/有备注的部分。



- 2021美赛心得分享-gwl_对小白友好的入门级介绍贴.pdf

  嗯，是咱学校（北理）21年F奖团队（当年F奖比例和O奖差不多，从22年开始F奖获奖率就从0.1%变成了2%，翻了20倍）的**小白友好**入门级介绍贴，**建议小白队伍一定一定一定要看这个！**
- 2406176.pdf

  俺们的**从始至终专注于解决问题**的论文！

00.2 一些不错的经验分享类干货🔗✍️

- https://zhuanlan.zhihu.com/p/446999082?utm_campaign=shareopn&utm_medium=social&utm_psn=1773375325663268864&utm_source=wechat_session

  一篇**小白友好**的美赛全流程介绍贴，回答了一些重复率很高的问题。

- https://levitate-qian.github.io/2020/06/18/MCM-experience-sharing/

  从第二个链接里摘出来的，听起来也是很厉害的o奖团队。是一篇**小白较友好**的美赛全流程介绍贴，较为细致的对O奖论文的架构进行了浅显的拆解。文章中提到的一些美赛评审标准也和我们队伍的认识很一致，如：

  > ### 6.4 5.4 美赛评审主要标准
  >
  > 1. **摘要**的要素是否全面，是否符合要求。
  > 2. 对**赛题的解读**是否正确，是否澄清了问题中可能的模糊概念。
  > 3. 是否清晰地列出了建模所需要的**前提条件和假设**，对其**合理性**是否给出了令人满意的分析与论证。
  > 4. 是否分析给出了**建模动机和对模型的合理性**做了详实的论证。
  > 5. 是否设计出了能**有效解决**问题的**模型**。
  > 6. 是否模型的**稳定性和敏感性**做了分析检验。
  > 7. 是否讨论了模型的**优缺点**，并给出了明确的**结论**。

- https://www.zhihu.com/zvideo/1465336604042285056

  一个**小白友好**的视频，对比h和o奖的区别。**论文手必须看**。我对这个视频的印象非常深刻，前两次打比赛看到这个视频的时候，感觉收获特别特别多。

- https://zhuanlan.zhihu.com/p/380305377?utm_campaign=shareopn&utm_medium=social&utm_psn=1773373526906753024&utm_source=wechat_session

  一个大满贯团队的**备赛历程&比赛过程**，**建议论文手看**，其次建议建模手看，编程手也可以来凑个热闹。里面有提到他们用到的一些漂亮图，如果你读完全文，你会**再次发现那个元老级别的模型总流程图的重要性。**

- https://zhuanlan.zhihu.com/p/370249046?utm_campaign=shareopn&utm_medium=social&utm_psn=1773374177497939968&utm_source=wechat_session

  一个以打模拟赛为核心的团队的**备赛过程**、比赛过程，以及论文手、建模手、编程手各自的**心得体会。建议论文手建模手编程手都看**，有助于加强本经验贴提到的一些观点的印象（如，论文手要积极搜论文、积极参与建模流程等）。阅读这篇经验贴时，就已经可以发现**O奖团队和写出来的论文的一些共性。**

∞. 假设

一篇数学建模论文不可能没有假设，所以这个经验贴也写一下假设，不奇怪吧。

- **假设1：**O奖表示着团队，至少看上去，在一切方面做到了非常好（不至于最好）。

  解释：否则这个经验贴将失去意义。

- **假设2：**O奖表示着团队的论文是一定符合O奖要求的。

  解释：否则本论文手在此分享的底气会变得十分不足。

- **假设3：**O奖的获得吃运气，但也存在可复制性。

  解释：评判论文本身就是个看眼缘的主观性强的事情，运气成分是一定存在的，讲究一个天时地利人和。同时，**O奖论文也一定有一些较为通用的共性，而达到这一共性的方法是可以被其他团队参考的。**

1. 数学建模比赛的本质⭐⭐⭐🤔

数学建模比赛的本质是什么？**解决问题。**那“数学建模”在数学建模比赛里是什么？**核心手段。**

于我而言，数学建模比赛中，**一个理想的数学建模流程**应该是（这里从建模的角度考虑，是因为写论文、求解模型之前，需要先确定模型。即，建模是第一步，是参赛成员打比赛、根据问题得出结论的核心手段）：

> 1. **确定选题**
>   - 通读ABCDEF六个题目（30min-1h）
>   - 讨论。结合编程手&建模手的能力储备 以及 题目内容，筛选掉一看就头疼的2-3个题目。（10min）
>   - 搜论文，对各个题目加深理解，大致确认解题的可行性（1h）
>   - 讨论。讨论题目的可解性，讨论找到了哪些可用论文甚至具体的模型和方法。
>   - 重复搜论文和讨论这两个环节。
>
>   第一天午饭前后应该确定选题，最晚第一天下午14:00需要确定选题。选题后不要换题。
> 2. **研究问题**
>   - 通读题目文档全文，而不仅仅是读小题。（30min）
>   - 讨论。在思维碰撞中确定每个小题的基本解决方向（可以是将题目归为预测/规划/分类/评价/优化等，更可以是大概确定模型的构建逻辑，**不要局限于将题目分类到传统模型类别中**），确定团队内部的理解一致。（15-30min）
>   - 搜论文。（1h）
>   - 论文手应该速读文章，将优质文章（最好能用一两句话说明为什么这篇论文好）发送给建模手和编程手。
>   - 建模手应精读论文的模型部分，思考是否符合题意。
>   - 编程手应精读论文的模型&求解，思考且仅思考模型是否可求解，没事干了再思考模型是否符合题意、继续找论文。
>   - 讨论。讨论各自的发现。（30min）
>   - 重复搜论文、讨论，直到团队确定一个**可求解的、合适的**模型（合适 = 不平庸、不落俗套）。
>
>   陌生概念的理解，和对基础模型的确定，同理。
> 3. **建立模型**
>
>   利用第2步研究问题的零碎成果（概念的定义、模型的确定），结合小题的具体情景，串联出团队该小题的解决方案。
>
> 对每一个小题，都进行第2、第3步。

绝大多数的队伍确定题目后的反应是，快速的浏览问题，根据关键词和设问，结合团队成员的模型储备，确定了模型种类，搜了搜论文但并没有找到现成的直接照搬就能用的模型或者代码，于是**在团队的模型储备中选定了一个模型**，接着进行编程求解。或者参照搜到的论文进行语文建模，用五花八门的方法得到结果。就这样，一篇h奖甚至m奖论文（当然，摘要写的不好就会喜提s奖），出现了！

但是：

- 你真的有读懂题目在问什么吗？
- 你是否明白了题目中的一切概念、专有名词？
- 模型储备中的那些基础模型，凭什么是解决问题的好模型？

请时刻提醒自己，数学建模仅仅是解决问题过程中的手段，而非目的。开始进行建模时，不要带着“这个小题我要选哪个模型”“那个小题哪个模型会比较有效”这样的想法去建模，有大方向确实是好的，但**不能沉浸在自己的模型储备里，不能凭借自己赛前的积累建立起对题目的认知。**正确的做法，是要提醒自己：

> 现有研究是否能够帮助我解决这个小题？

当你回顾美赛、国赛历年题目，你会发现绝大多数题目（甚至全部）都是来源于现有研究，或者题目是现有研究在某方面的局限（美赛2019A题，建模的对象：龙，就是对大型哺乳动物的一种局限）。

因此，为了更好的解决每一个小题，需要在建模之前，花大功夫去搜论文。近五年的现有sci高影响因子论文（反正最好是一区二区的）是对该问题的最前沿研究，是最值得信任、最有助于解决问题的。并且，sci高影响因子论文对基础模型建立的帮助，非常之大。我们的论文基础模型（RTWP模型，Real Time Win Probability）就是直接利用了现有论文的模型，帮助我们解决了最关键的环节——在一场正在进行的比赛过程中，计算网球运动员的实时胜率。

到此，阐述了一个最关键的观点，即模型应源于现有研究（源于搜到的论文）。团队应该站在高大巨人的肩膀上再进一步，而非自己造一个（大概率矮矮的）巨人。让我们回头再看前两个“但是”。首先，开始搜论文之前乃至团队开始讨论之前，务必通读问题全文，并且标清楚每个小题的关键词和每个小题自己不清楚的、需要团队进行定义的术语。比如2024年C题，全文最关键最核心最重要最...决定团队上限的概念阐释，当属对“momentum”的阐释。momentum，直译为动量/势头，需要被团队进行量化。对momentum的量化方法，对于每一个团队都是陌生的。每当团队遇到了“陌生”的事物，请立刻提醒自己：

> 查论文！查论文！查！论！文！！！

是的，当时我们对momentum的量化也很苦恼，于是我们就用诸如`tennis,momentum,quantification`这样的关键词来搜论文。当我点开那篇1985年统计心理学论文（很尴尬的是我们忘记在reference里引用这篇，不过我在论文第一部分introduction的第二段就加粗写了这篇论文的作者和发表年份），我就像在漆黑的洞穴里找到手电筒一样兴奋。文中对momentum下了非常明确的定义，并且定义和实时胜率相关。至此，我们能够将基础模型（RTWP模型）使用小学二年级学生都会使用的减法，和赛前胜率挂钩，得到核心模型（RTMQ模型，Real Time Momentum Quantification）。

STOP！屏幕前聪明的你也许发现了尚未被完全解释的概念！——赛前胜率。没错，这又是陌生事物，于是我们又去搜论文，发现一个如今应用广泛的模型能够计算选手的赛前胜率：Elo rating system。当然，复现这个模型的过程中又遇到了不少陌生的参数，有的查论文搞定，有的搜数据集搞定，...

发现了吗？我们模型的建立，就是在一步步定义未定义的概念，一步步确定未知的参量，一点一点解决团队面临的问题。有人会说，这难道不是所有队伍都会经历的吗？但是，有多少队伍能够无时无刻从“解决问题”的角度展开工作？从解决问题的角度进行建模看起来很费劲，因为一切都需要团队自己检索。但是实际上，这是最高效的办法，因为这种方法基于已有成果，而已有成果（大多数）必定是前人研究后发现可行的模型/方法，或者直接就是前人的研究结果。从“数学建模”的角度进行建模，会在第一时间将团队的思绪局限在那些耳熟能详的数学模型：规划，预测，评价，具体而言也许就是微分方程，马尔科夫链，元胞自动机，模拟退火...

面对陌生事物，万万不要拿着半杯子旧水对其进行研究，而是应该老实把这半杯旧水咽下去（模型储备不要在基础模型建立的早期过多使用，甚至在早期不要使用。当然，如果你真的真的很熟悉，那就可以去用，或者**这个小题是论文核心模型的延伸且现有研究处于空白，那就凭通用经验大胆逻辑推理出合理的模型即可**，**核心模型【即团队的主要模型】必须从研究问题开始慢慢搭建**），用一个空杯子重新开始接水，从零开始、从问题的底层（明确团队为了解题应该完成的任务、熟悉题目中的陌生概念）开始，建立对问题的最优质的认识（搜论文，将有用者为我所用），**进而用一杯新水，解决问题。**

解决问题嘛，当然需要先完完全全的了解问题是什么，再优先参考别人都是怎么做的，进而得到我应该怎么做才最能解决这个问题。

还有一个很重要的平衡是需要经验积累的，即研究问题到什么地步再开始第3步真正建模。研究的太过（论文找的太多，解决路径过多，解决方法过于复杂）会导致建模过于复杂，反而不利于解决问题。所以这个度，需要掌握。不同的角色（论文手，建模手，编程手）对这一平衡的把握有差别：

- 论文手从写论文的角度，想要一个传统模型之外的、模型建立过程尽可能详细的模型，避免论文平庸。
- 建模手从建模的角度，想要一个具象的、公式够多的、模型结果讨论充分的模型。
- 编程手，只需要一个能实现（**的基础上**尽可能高大上）的模型，不然嘞？

因此，第2步（**研究问题**）这一行为的截止，主要需要从编程和建模两个角度进行考虑，因为编程实现是数学建模比赛里最关键的环节。一个优秀的论文手应该也必须能够驾驭一切模型的语言描述（再不济上gpt/newbing啊）。

1.1 不要把数学建模当作本质的原因之二

我看了今年C题o奖论文，大概总结出了大多数队伍打美赛的方法。大多数队伍并不是以一个做研究的角度出发打美赛，而是真的发挥团队的建模能力，拼美工、拼...(没总结出来)。这其实也没有任何问题，但从得奖的角度来说，评委看到的论文99%都属于拼建模的论文。其一，论文的创新性会下降。其二，凭什么A队伍的论文比B队伍的好，明明用的模型很类似？所以，比拼建模，不可避免地会让团队和其他队伍出现较大程度的竞争，也就是需要卷美工，卷...(依然没总结出来)。所以，我非常推崇从研究问题的角度解题，而非从数学建模的角度解题。也许，因为我们并非采用了常规的建模方法，我们的创新程度比我们提交论文时想象的更好。我可以非常草率的对比一下两种打比赛方法对团队能力的考验（仅代表个人看法，⭐越多表示要求越高）：

	通过研究问题打美赛	通过数学建模打美赛
文献检索能力	⭐⭐⭐⭐⭐	⭐⭐⭐
建模手建模能力	⭐⭐⭐⭐	⭐⭐⭐⭐⭐
美工能力	⭐⭐⭐	⭐⭐⭐⭐⭐
编程手编程能力	⭐⭐⭐⭐	⭐⭐⭐
论文手逻辑能力	⭐⭐⭐⭐⭐	⭐⭐⭐⭐

2. 美赛到底考察什么？

2.1 核心

正是解决问题。金模型，银模型，不如目前最能解决题目问题的草模型。当然嘞，最能解决题目问题的一般都是钻石模型（科研论文成果）。

2.2 美术建模？ ⭐🧐

2024了，还有多少人在成天吐槽美赛的美术建模？？一句美术建模可以是对于部分团队专攻美工的吐槽，也可以是对于自己团队失利的不满，这些话用来开玩笑都没有问题，怕就怕哥哥姐姐弟弟妹妹们当了真。

首先，一篇论文的美观程度天然会增加评委的好感，人类天性爱美，这是毋庸置疑的。但是，这不意味着所有团队都需要非常认真努力的学习如何画出漂亮的图。我们团队使用的所有图片都是ppt/excel/python绘制，需要手工绘制而非代码生成的图大都是我用ppt画出来的，并没有太多花里胡哨的操作。当然，不可否认，有许多在线网站/专用软件可以画出更美观的图，我的建议是，如果团队之前会用，那就用。如果之前不会用、想在备赛时学，那就量力而行的学，**不要把搞美工当作美赛的核心。**

在我眼里，画图只有一个金标准：

> 能够充分的表达模型结果 或 能够充分表达团队想要表达的含义。在此基础上，尽量又快又好的画图。

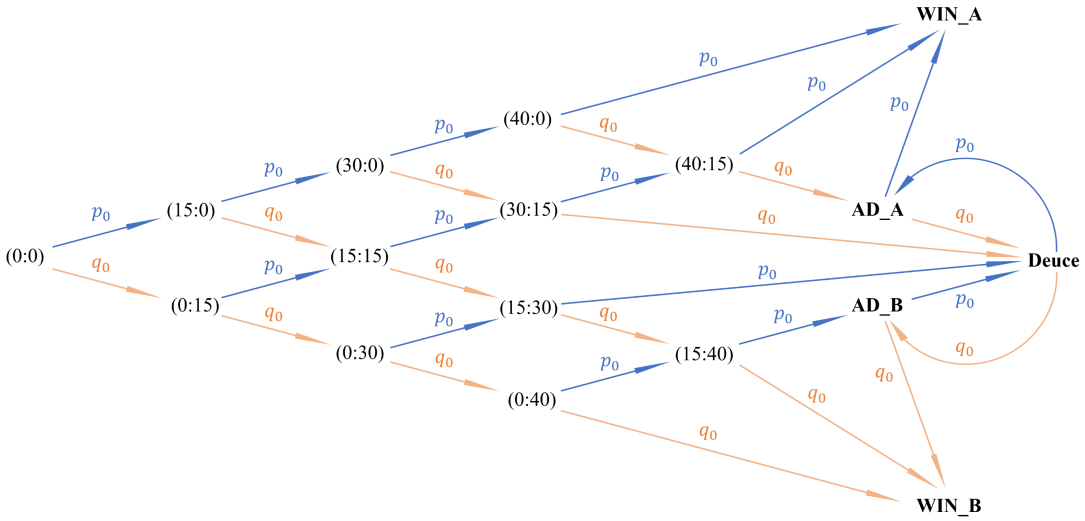
拆开讲解一下这个标准：

- 能够充分的**表达模型结果** 或 能够充分**表达团队想要表达的含义**

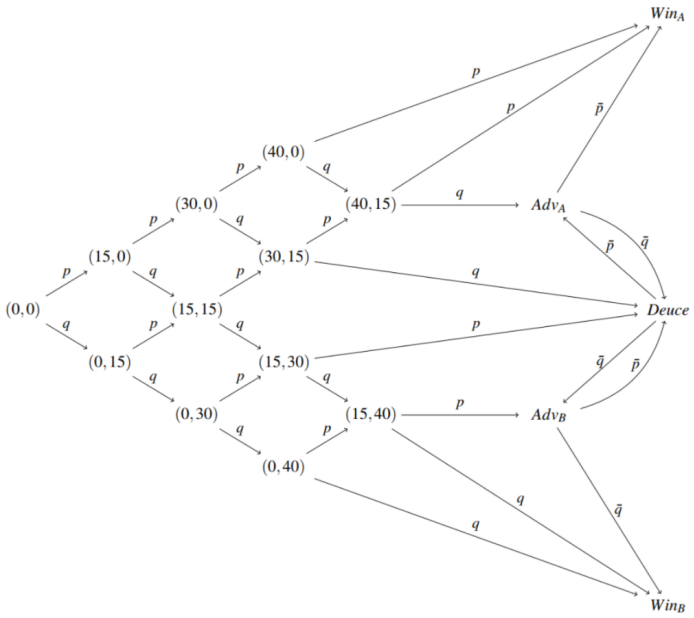
  这个很好理解，形式为内容服务。这一点大家基本都清楚，包括但不限于折线表达趋势，饼图表达比例，雷达图表达多维度对比，等等等等。不清楚哪种图合适？搜一搜、再想一想嘛。

- **又快又好**的画图
  - 快：团队中的论文手应该在正式比赛之前的**备赛阶段，储备一些底图**（如ppt画好introduction部分的模型流程图的框架，这个必须事先画好，所有O奖论文必须有模型流程图）。最简单的办法是照着你觉得漂亮的图，自己画一个，这样你就有很多个各种各样的漂亮的图了。等开赛的时候，把内容改改基本就行，不用太注意形状之间的对齐精度。实在是有时间的话，再把画图这件事情从done做到perfect。

- 好：从往年O奖论文里摘到的漂亮图，就是所谓的“好”。当然，自己有艺术天赋，非常鼓励自己创作，比如下图就是我画的（叉腰，利用独立同分布假设，将网球竞赛任意比分任意一方选手的胜率量化出来，用于计算选手的赛前胜率。这个图的优点在于，首先这是从别的论文里看到的图，我觉得不错，就复现了一下，画的肯定比原来的好看。其次用了蓝橙配色，颜色全部是ppt里自带的色阶（这种事情能偷懒就偷懒，好不好？），较为和谐；然后意义明确。



参考原图：



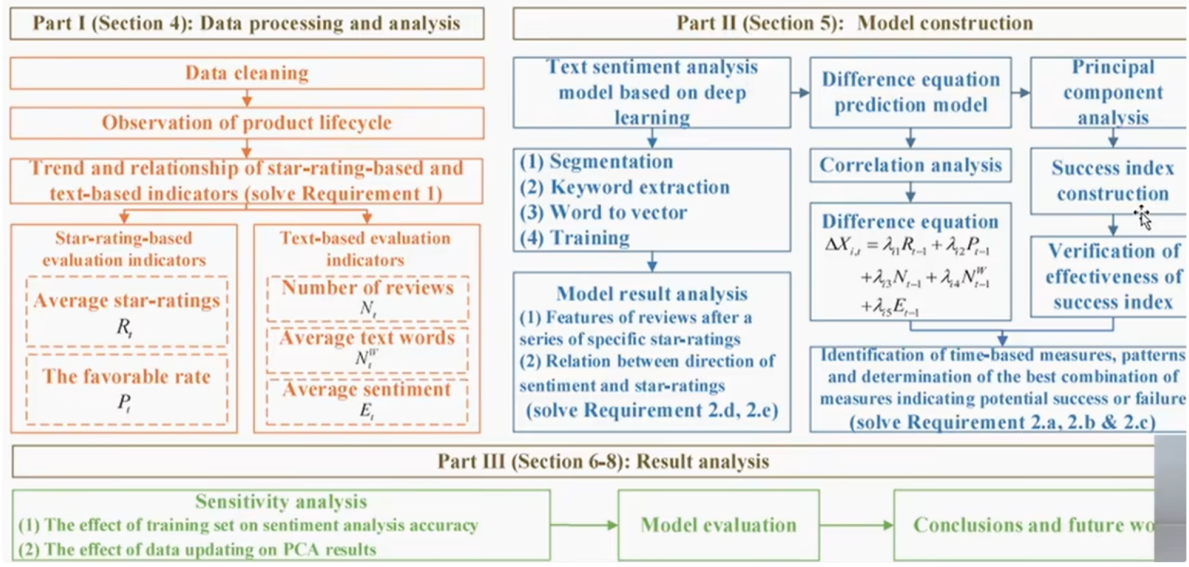
咱就是说，也不知道为啥作者要画个小飞机（像轰炸机）的形状，这是打网球又不是开飞机。。。

ok，都说到这了，简单分享一下我眼中必须要有的模型总流程图，以及我自己绘制的一个特别特别好的核心模型流程图。

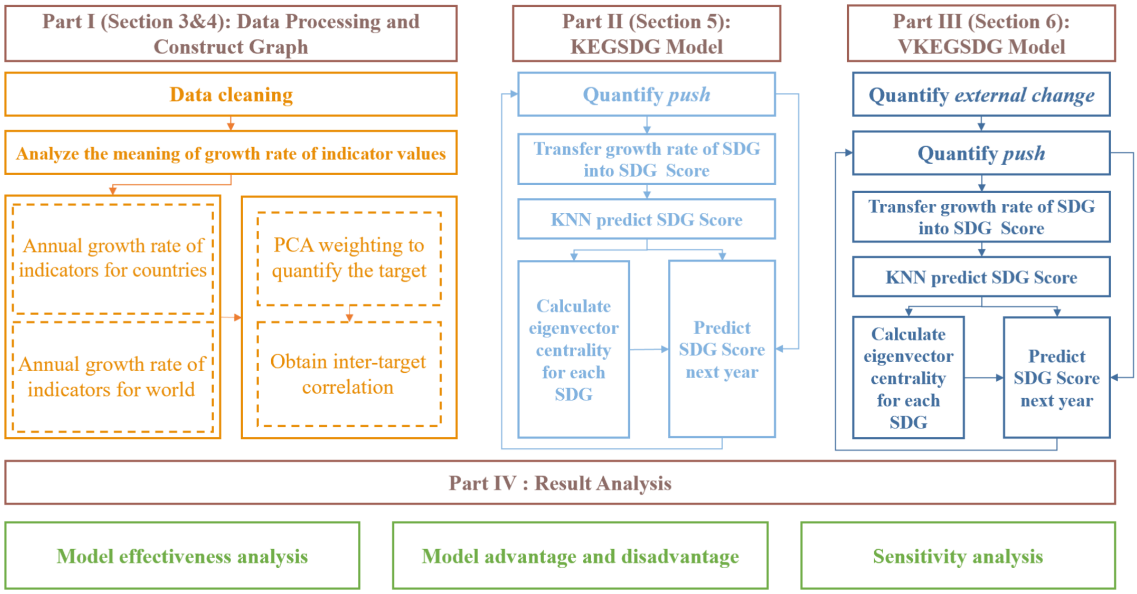
2.2.1 模型总流程图

必须有，必须有，必须有，重要的事情说三遍。从五六年前的一篇元老级o奖文章开始，大批的o奖文章都使用了完全一致的排版甚至配色（没错，我们也是用了一样的配色）。

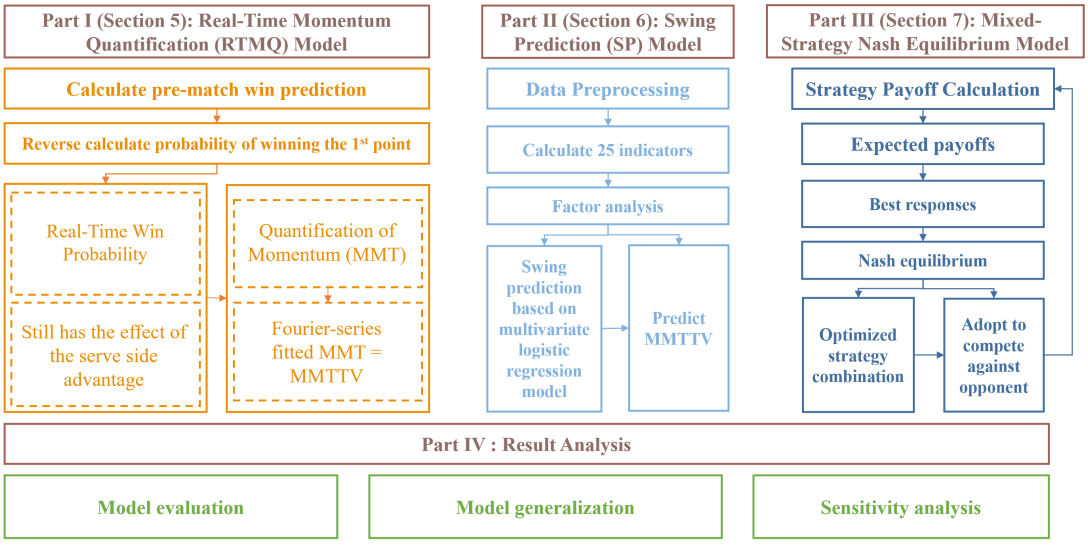
原图长这样：



我2023的H奖论文长这样：



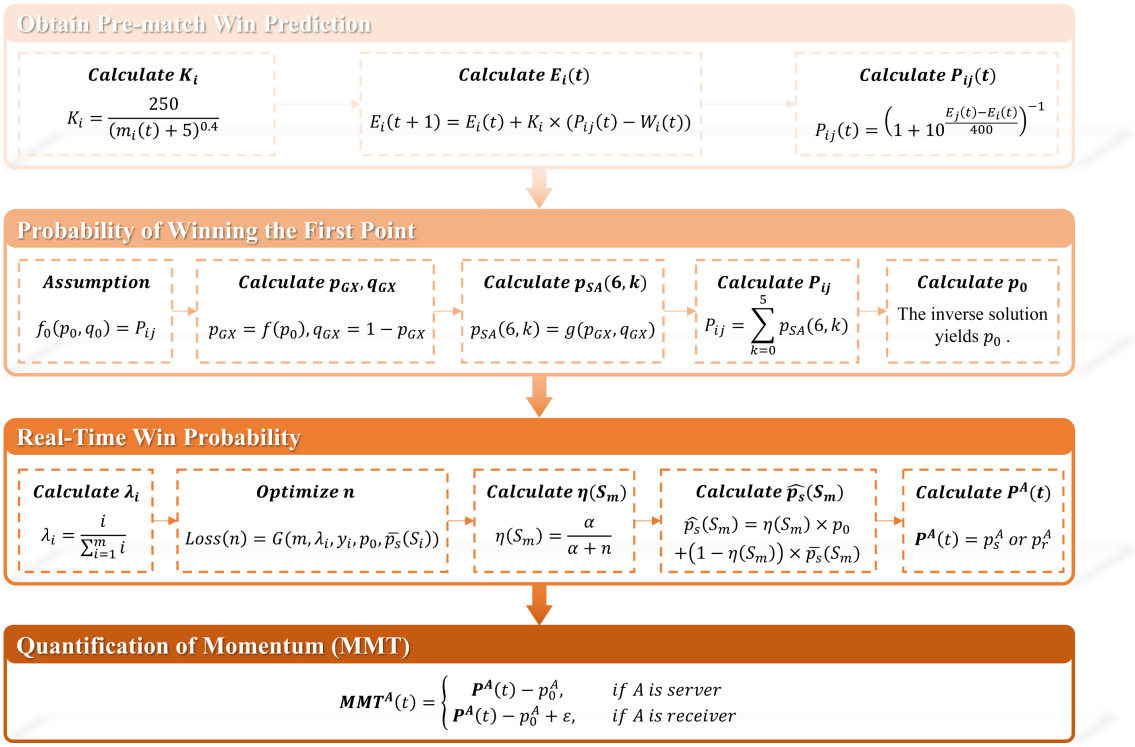
我们2024的O奖论文长这样：（别问我为什么比去年的丑，今年的是我10分钟画出来的，甚至差点忘了画...所以大家一定要留出足够时间来做收尾工作，不然就会着急忙慌，大大增加忙中出错的可能，比如没写标题...）



照葫芦画瓢？这简直是照葫芦画葫芦啊！人家是换汤不换药，我是连汤也不换。当然，最好还是要把关键符号给写上去的，不然相比元老图，还是差点意思。

2.2.2 核心模型流程图⭐🤩

这个图可以说是我**五次数学建模比赛里画的最满意的一张图**（RTMQ模型建立过程）。可复制性极强。



优点包括

- 让评委轻松愉快地**抓住建模的主线**，一个图就能概括5-8页的论文

  而且，模型建立流程越复杂，这个图的含金量越高。我反正是越看越满意，不知道评委是怎么想的。
- 我觉得其他o奖论文的图中，好看的没我们逻辑清晰展示简洁，逻辑清晰展示简洁的没我们好看。
- 有助于后续写摘要时，用最简洁的语言描述模型的建立过程

  当然，每个模型内部的流程的细节是不必要写的。**摘要写作方法见本文4.3**。

2.2.3 但是...🤯

如果团队经过研究问题，兜兜转转回到了传统模型（几乎所有赛题都可以用传统模型解决，不过有的队伍得了S，有的队伍得了O），那就真的需要多看看如何呈现数据最漂亮。我在24年o奖论文合集可以获取之后，看了看C题其他队伍的论文（出乎意料的是还有其他队伍没写标题哈哈哈哈），发现大多数队伍都还是使用了传统模型，但美术做的真的是**花里胡哨**，我看着挺头疼的。我觉得，只有一种情况需要团队做好美工：

> 建模手/编程手能力很强，团队以建模手/编程手为主导，且这两个人中至少有一个人能够写模型建立和结果（其实这很简单）。此时，这两个人应该去招一个尽量有经验的+会做美工的+有尽可能多的o奖论文漂亮图储备的（这可以备赛期间整）+英语能力强（帮助团队判断表达是否地道）的“论文”手。

这种情况下，“论文”手的主要角色是团队的美工咨询师，应该在所有情况下都给出团队有关图类型/细节的建议，包括但不限于模型流程图、五花八门各种各样百花齐放花里胡哨的解释说明图、模型结果图（也就是python/matlab跑出来的图，照样是可以美化的）等等。

2.3 创新性? ⭐

美赛很关注创新性，有很多博客都说美赛重创新、国赛重结果准确性，但是大多数队伍不太能做好创新这一点。我猜测很多队伍都会倾向于认为创新性是团队成员发散思维后进行的天马行空/有理有据的**猜想**，因为我曾经陷入了这个误区。如今，个人认为，论文创新性的最大来源就是，**不生搬硬套传统模型**，而是通过阅读文献、检索数据，系统地**研究问题**来确定一套解决方案（即确定团队要使用的模型）。因此，**我倾向于将创新性当作解决问题路上的副产物**，而非团队需要努力达到的一个目标。即，不要为了创新而创新，应该为了解决问题而“顺路”创新。再次重复，个人认为数学建模比赛的核心是解决问题。

2.4 可读性

玄之又玄，可归为主观评价或运气成分，但也有点说道（连贯性、流畅度、逻辑性）。O奖论文一般分为两种，一种是可读性极强（解释充分），另一种是极度硬核（真正擅长建模的队伍更可能使用硬核的语言）。每次比赛我都会选择前者。

> 一篇数学建模论文可读性之强，**形式**上在于建模过程中对公式、结果的**充分的讨论与解释**，**核心**在于**用语言构建紧密的逻辑链**。并且，论文手的表述越**结合题目**、越**具体**，可读性越强。

这个确实比较吃天赋，因为人们日常表述习惯不同，能够在大多数情况下保证自己的“谨言慎行”的人更少。不严谨的、逻辑跳跃的语言，所描述的事物是欠定义的、有多种可能的，只有严谨、具象、逻辑链完善的语言，才能清楚地描述出真正想要描述物体。hmmm...说着就抽象了起来，多看O奖论文熟悉一下表述吧，我觉得我们的论文写的就很清楚，可读性也不错。

对于美赛而言，论文的连贯性其实非常重要。即，大多数情况下，论文手应该用一个个文段进行表述，而非不考虑场景地使用各种奇怪的表述形式，包括但不限于

- 莫名其妙地开始进行条列
- 没错就像这样，这个叫做**无序列表**
- 如果观点之间不是并列的，就不要考虑使用无序列表进行表述的分块化，这严重破坏了论文的整体性
- 就很不好，别学。

或者

1. 这个叫做**有序列表**
2. 我没有见过美赛有论文这么用的，所有就不要碰这个有序列表了
3. 嗯嗯，真没见过。。

等等表述形式。文段，永远是最好的方式。（我们的论文这点做的比较好，可以观察下是否全文大篇幅地使用最简单的文段进行表述、无序列表的使用是否合理）

2.4.1 9个操作性极强的小技巧✍️

- 用gpt对团队论文进行润色

  From Chinglish to English.

- 不要滥用无序列表或有序列表
- 全文（特别是**摘要**！！！）一定不能多次使用同一种句式，**句式一定要多样。**

  这种事情，如果论文手自己没把握，就请交给让人放心的gpt小哥吧（我80%情况下都会参考gpt润色的结果）。可以看看我们的摘要，长难句基本都需要参考deepl的翻译和gpt的润色，否则自己写的根本没法看。

- 每个模型的最开始，对模型进行简单介绍，并简练地复述建模过程。
- 每个公式都必须结合问题的具体情境进行解释。

  如，每个参数在问题中的具体含义必须解释清楚。

- 对题目中非常关键的概念、题目特别标注需要团队注意的、涉及题目核心的，进行加粗。

  核心目的是方便评委**一眼看到我们如何解决的问题**。这种时候，方便别人，就是方便自己。当然，不要滥用加粗的强调方式。

- 论文的强调内容，进行**加粗**足矣，不要搞花里胡哨的强调格式（🈲换颜色）。
- 每个section之间适当地使用一段话进行承上启下。

  这对于增加论文的**连贯性**（可读性）非常有帮助！

- 对于公式过多的模型或算法，请善用**伪代码**。

  我这次比赛一口气用了四个伪代码。其好处在于让评委看了一段时间公式之后就能看到伪代码，**平衡**论文里建模部分的**不同表述形式**（文段描述，写公式，写伪代码，...）的比重。当然这里不是说用多种表述形式表达同一内容（闲得蛋疼呢），而是用多种形式表达不同内容。

3. 为什么我们拿了O奖？有哪些优点可复制性极强？

讨论完数学建模比赛的本质以及美赛的考察点，我可以粗略归纳出我们能够拿到o奖的几个原因。

3.0 相信自己，谨慎参考付费思路⭐😊

请记住，只要团队进行了`研究问题`，**团队对题目的认知和建立的模型一定要比网上任何一份（付费）思路要好。**网上（付费）思路的存在意义，是团队研究问题后发现没什么思路时，可以提供给团队一些参考。一定不要认为网上的思路都是大神写出来的“正确答案”，首先美赛没有正确答案，其次大神为了快速抢占市场一定不会很细致地进行研究问题这一环节。当时，我们的思路区别于任何一种网上的思路（我经常在微博知乎搜最新的思路，作为参考），但由于我确定我们`研究问题`的过程是逻辑严密的，我选择忽视了所有的思路。
思路的最好来源就是，搜论文！搜论文！搜论文！！！

3.1 语言表达⭐

可能有很多F，M甚至H S奖的论文都能够在建模上比肩或者超过O奖论文。但是O奖论文可贵之处除了其解题思路，就是**语言一定是被评委喜欢的。**这一点我觉得我这个论文手达到了评委的预期。

- 真的真的要**确保不要说chinglish**，只要对自己英语水平有一丁点的不自信，写出来的论文就很容易会让别人一看就是中文表达，因此一定一定要**让gpt帮忙润色。**
- 同时也要**减少使用第一人称进行口语化的表达/解释。**总之，一切都是为了论文的观感好。

毕竟，亲爱的论文手，你也不想让自己的论文因为表达扣分而从O变成F、F变M、M变H吧？

至于语言表达能力如何培养，请看往年O奖论文，这是最好的培养办法，也是捷径。**O奖诞生于模仿。**

3.2 搜论文⭐

这也是我们论文最可贵的一点，正是因为我们花了足够多功夫搜论文，研究问题，我们才能写出如此完善成熟的解决方案。希望各位看完这篇分享贴后，**搜论文、查资料**的意识根植于心。

3.3 扎实的建模（概念及应用）、编程基础

建模方面，csq在寒假非常认真的学习了大部分的模型，她在我们定题后开始找论文的第一时间就在知网上搜到了利用博弈论和纳什均衡计算网球竞技中策略选择的一篇论文，当时她就说博弈论很可能是C题后面的小问需要用到的。这就是一位优秀建模手的**直觉**，能够最快给团队一个大方向，不让团队的思路跑偏。建模手的建模环节一定不会涉及过于复杂的手算，但需要建模手保持一个清晰的大脑，清楚自己正在做的任务是为了解决问题中的什么问题。建模手的另一个重要之处在于需要和编程手进行工作交接，建模手的建模结果（通常是一个流程，每一步流程包含完全定义的、清晰的数学公式和所有参数的取值、变量都是哪几个，等等等等）需要交给编程手进行实现，所以建模手也必须能够有清晰表达建模过程的能力。

编程方面，瑞祥哥自身就是计科保研学长，熟悉大数据和图论，因此其编程能力很强，非常让队友放心。但是这不意味着他在比赛过程中手搓一切代码。相反，他几乎很少动手敲代码，而是使用**Github copilot**（参考链接：Github Copilot 结合python的使用 - justtest1 - 博客园 (cnblogs.com)）进行辅助编程。由此，团队编程方面的生产力非常富裕，可以说是只要有了建模结果就能在最长两小时内得到一切我们想要的结果。当然，编程手有任何实现方面的阻碍，建模手都应该立刻进行解释或进行补充建模，确保能够实现。

3.4 清晰的团队分工

因为我也并不清楚其他队伍都是如何分工的，我仅以我们为例进行介绍。

我是队长（负责把控整体思路和团队进展）&论文手（同时深入参与建模流程，建模手有困难时及时下场帮助推进），csq是建模手（此外帮我写了第八部分的论文+灵敏度分析，确实是论文手每次比赛最后的时间都太紧了，需要分一些相对容易的论文任务给建模手），瑞祥哥是编程手。这其实也是我心目中最理想的配置，即论文手一定要熟悉建模，并且一定要很有逻辑，这样才能把控好论文的整体思路。然后建模手和编程手都是在论文手敲定总体思路后（当然，之前需要一起讨论出一些合理选项，并且一起做决策）各自处理各自的任务，到了任务节点（ddl）论文手就来看一下进展。总之就是论文手进行合理的任务分配和人员调度，然后建模手和编程手配合。

当然，队长的身份可以落在任意成员的身上。之所以说最好论文手担任队长，是因为评委直接观看的是论文，在评奖这一块论文手起到的**写论文的作用**一定是大于建模手和编程手的（不过，如果建模手非常厉害，把建模部分的论文给写出一个底稿，那也不是不可以）。**得到F、O奖这件事情上，虽然谁都不能不靠谱，但论文手最不能不靠谱。**

队长的意义在于让其他成员安心处理自己的任务，减少浪费在队内辩论上的时间，但同时每一个决策都必须经其他成员点头才可以敲定下一步的任务。总之，leader的意义还是蛮大的，相应的责任也很大。好在，我们三个人任意拿出一个都非常让人放心，所以此时leader的核心功能就是整合自己、编程大神和建模大神的思路，给出自己的优缺点判断，进行短暂的利弊权衡之后，帮助团队敲定最有助于解决问题的方案，然后进行快速的分工。

这只是一个参考，我相信每个队伍都会有一个自己舒适的分工方案。

3.5 时间安排及进度把控

我们每天都是8:00-8:30起床，中午该吃饭吃饭，想午睡就午睡，保证三个人的精神状态是良好在线的。每天23:30-次日0:30收工，保证睡眠（我偶尔会收工后再搜搜未解决问题的解决思路，但还是优先保证自己的休息）

我每次比赛都会对时间安排和进度进行把控，比如核心模型（往往是前两问）的结果必须在建模比赛过半前（第三天中午，最好第二天晚上就可以）搞定，**比赛结束前必须留出整8个小时进行论文的最后优化**，等等等等。基本上每半天都要进行一次进度安排的调整（松弛有度）。

一定一定不要在比赛的前期过于懈怠！不然后期就会草草收尾，甚至无法完成论文，缺少后面小题的模型。

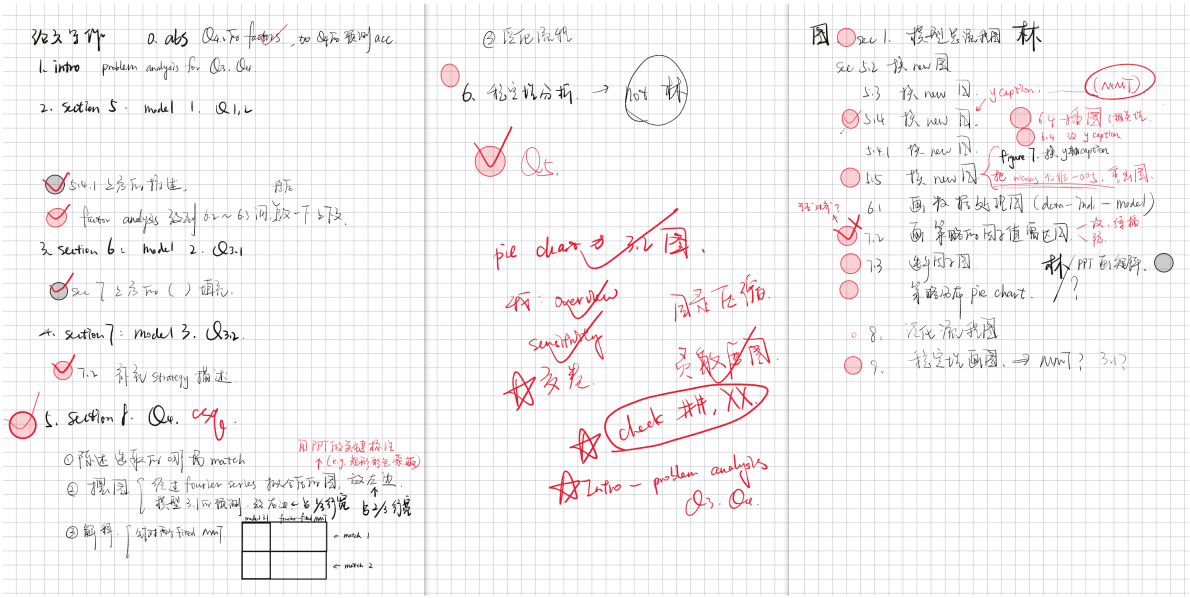
另外，论文手一定要和其他两人一起讨论模型。一个团队里，建模手可以对模型的细节最熟悉，但是论文收必须对整体建模流程、模型之间的联系、以及为什么这个模型能解决这个问题，有最深入的了解和记忆。论文手必须能够条理清晰地阐述每一个模型使用的原因，还有团队使用的每一个解决方法的原因（比如，为什么改这个参数就可以实现不同模型的对比）。确定模型、建模手和编程手开工后，论文手才应该开始写自己能写的部分，而不是从定题后就开始不管建模手和编程手进行写作。

最后优化清单

我往往习惯于带着整个团队在最后八个小时完成以下内容：

- 写作部分
  - 摘要
  - 第一部分的background information
  - 稳定性分析
  - 美赛的最后一个小题 (往往是写一封信或者邮件等等，注意格式)
  - 参考文献
  - 附录（若有必要。请注意，美赛不必在附录内贴代码，这是浪费写作空间的行为）
  - 压缩目录（三个人都不会且没时间找资料现学了，所以我们的目录很丑，应该写在一页里的）
- 模型结果图部分
  - 挑一张元素最全的图片，大家一起确定好所有内容的规格，包括颜色、字体、字号、线条粗细等等。之后让编程手把需要迭代的图按照标准全部重跑一遍，论文手负责粘贴进论文里
- 其他任何零碎的事情

我们当时的优化清单大概长这样，反正不管写得多乱（毕竟时间紧张）有一个人看得明白就行：



3.6 论文框架 ⭐ ⭐ ⭐

取得好奖项的论文大都不约而同地采取如下基本框架和写法。请注意，学习论文框架最有效的方式依然是找一篇得了最多奖的O奖文章，一段一段地看该篇论文在写什么。如果感到总结段落大意比较困难，请借助chatgpt或者newbing。chatgpt现在已经具备读文件的功能，newbing则是一直可以在edge或者edge dev浏览器中打开pdf格式文件时在侧栏中使用。

- 标题

  （还是要写的。
- 摘要
- 关键词
- 目录

然后，是正文部分：

1. 介绍 Introduction
   - 1.1 背景信息 Background Information
     - 第一句话进行引入。
     - 接着，点明主要研究对象。
     - 最后，一两句话带过全文的研究内容。
   - 1.2 问题分析 Problem Analysis

     逐个小题进行**重述/分析**（必须重述，分析是可有可无的）。O奖论文在这里会有两种写作形式。一种是，针对每一个小题，重述原题中的表述，采用地较为普遍；另一种是，在此基础上，写下团队的思路(我们就是这么写的)。第二种缺点在于浪费写作空间，但优点在于再次让评委明白团队每一问的思路（即，将问题分析视作摘要的“细节更多版”），提到更多涉及解题关键的细节。
   - 1.3 模型概览 Model Overview

     这里放**模型总流程图**。发挥论文手漂亮图储备的时候，到了！
2. 假设 Assumptions

   把团队解题过程中最关键最核心的2~5条假设写在此处，**通常这些假设都和模型的选择有高度关联。**
3. 符号 Notifications

   把团队解题过程中最关键的一些符号含义（如核心模型里使用到的。**并不是所有符号都需要在第三部分写下来，但每一个符号必须在公式后的文字解释中有精简明了的解释。**）进行阐明。
4. 数据来源 Data Source

   如果论文中用到了团队自行搜索到的数据库中的数据，需要有这一部分汇总团队使用的数据库和目的。
5. 模型名称 XXXXXX

从此处开始就是最核心的建模流程。对于每一个模型，都可以按照如下步骤来写（不必区分小标题，建议小标题的设立和团队对小题的解答有关，或者和建模流程有关。如果团队正在写的部分和原题中某个小题的解答有紧密关联，建议设一个小标题，或者到建模流程里的下一步骤了，也可以设一个小标题。否则到下一个模型之前都不必要写小标题）：

- 简介，即介绍为什么使用这个模型。陌生概念的阐述与定义应该在这部分进行。
- （如果该模型是核心模型，还应该写建模流程、画核心模型流程图）
- 模型建立流程。一个公式，一段解释。再一个公式，再一段解释。如果公式太多或者模型太简单，可以运用伪代码来丰富建模过程的表达形式。
- 模型结果图，和充分的、扣题的解释。

6. 模型优缺点分析及未来工作 Model Evaluation

   这部分比较灵活，标题的表述也很灵活，建议在全部模型写完之后在新的section（新起一个最高层级来写，会latex应该明白我的意思）中写。不需要写多，客观展现即可。

7. 稳定性分析 Sensitivity Analysis

   必须有，即一定程度地改变某个关键，参数观察模型结果。一段解释+一个图即可。

8. 参考文献 References

9. 附录 Appendix

   真有必要再写（比如过多的公式、过多的结果图，等等），否则没必要写。

10. 信/memo/。。。

    即，回答美赛的最后一个小题。不用写的多漂亮，但依然建议挑选一个好看的模板写。word写作更方便，可以word写完后转pdf，然后和latex导出的pdf进行合并。具体方法没实践过，我们写的memo很丑。

4. 一个优秀的论文手应该做到哪几点?

4.0 语言表达

简而言之，论文里要**书面化表达而非口语化表达**，此外要**说english而不是说chinglish**...总不能让评委一眼就看到这是中国学生写的蹩脚英语。这很大程度上是O奖筛选出来的原因之一。想要冲O，就不要忽视这一点。如果英语能力有限，活用chatgpt吧，这东西不会辜负你，但请学会如何写提示词。

写提示词的基本规则 ⭐⌨️💡

不知道为什么就写到了这么详细的点。。简单介绍一下我一般比较常用的写法（针对论文润色）：

- 明确gpt的**身份**。

  如，一位有着500年论文润色经验的教授。

- 清晰阐述你的**目的**。

  如，将下面一段文字用英文表达（当然，你给gpt什么语言的内容都行，只需要明确它生成必须是英文即可）。

- 给出待优化的**内容**。
- 清晰阐述你希望生成内容的**风格**。

  如，不能遗漏原文段里任何的细节 / 若非常有必要可以对冗余or重复的内容进行删减or整合、使用书面化的表达、使用学术论文一样的严谨且有逻辑的风格。

现在写提示词已经发展成了一个专门的工作，干活的人叫“提示词工程师”。本质上是提出足够清晰具体的需求，从而让大语言模型发挥出更出色的实力、使用更多的模型储备，等等。网上有大量内容讲述如何写出优质的提示词，我给的方案一定不是最好的（当然，已经够用了），这个小经验贴便不过多赘述。

4.1 论文逻辑性、连贯性

前文3.1已经描述过，注重表达的连贯性，要让评委读着像一篇文章（雅思里写作的评分依据有一条特别适合表述这里的意思，就是`coherence and cohesion`）。句式也要丰富，总之看2.4.1。数学建模比赛中，对全文的逻辑、模型最熟悉的，不应该是建模手，而应该是论文手。为了保证表达的流畅，论文手必须时刻明白小题之间的联系，以及小题内部模型建立的流程。建模手长期浸淫于模型细节，反而可能容易忽视大局，这恰好是论文手必须要时刻注意的。

4.2 一个合理的写作顺序

我习惯的写作顺序如下（只说最关键的内容。符号，数据库等等很容易就写好的东西都不在此提）：

1. 写`2. Assumptions`

   这个是在研究问题的过程中就可以写的，用零碎时间就可以写完。

2. 写`1.2 Problem Analysis`中，第一问的问题分析.

   这是因为，论文手开始干活的时间点是在讨论好第一问模型之后的，这个时候论文手能写的就是`1.1 Background Information`、`1.2 Problem Analysis`和第一问模型。考虑到论文收应该先理清自己逻辑再开始写建模详细流程，建议论文手先写问题分析，顺带着就可以理清思路。当然，写完建模过程之后，也要回头看一下是否有和自己的建模过程写的冲突的地方，有冲突的要依据建模流程改问题分析。

3. 写`第一问模型`。

   不然回头去写背景信息嘛？背景信息很好写（其目的就是引人入胜，最后用一句话介绍文章是关于什么的，让评委有个记忆点），可以往后放放。写第一问模型的过程中，就要开始同步画图了。

4. 对每一个模型重复第2和第3步。

5. 写`Abstract`&`Sensitivity Analysis`。

   如果来不及，稳定性分析让编程手/建模手写。

6. 写`1.1 Background Information`

4.3 会写摘要...但这并不难！ ⭐⭐⭐🔍

大家都说摘要很重要。摘要确实重要，就像是你看帅哥美女，除了看气质（粗略浏览论文排版）以外，下意识映入眼帘的就是ta的面庞（位于论文第一页的摘要）。我一直觉得摘要写作很简单，因为全是套路，没太多诚意，大家面对写摘要这件事也千万不用太犯怵，真诚（真诚 = 可读性强）的地方全在论文正文。

一定要保证摘要页（包括**标题**和关键词哈。。。。😓）在一页以内，不要超页数。一定要亲自去看O奖论文的表达，以下是我写摘要的通用思路：

> - 摘要的第一段：第一句话说清楚背景和研究对象，后面两三句话话概述一遍文章建立了什么模型（不用说所有的，说最关键最核心的最大的一两个模型）、都解决了什么问题（第一段所提到的每个模型都要说建立了什么模型、解决了什么问题，没提到的模型当然就不用说）。
> - 摘要的第2~n段：主题段落，即讲解每一个模型都是如何建立的。每一段的写作思路都一样。讲解每一个模型建立过程中的核心步骤，包括使用的具体**模型名称（加粗强调）**（必要时，也可以说一句建立该模型的目的，或团队认为的**有关解题关键的不得不说的说了会加分的细节**，但优先保证摘要的精简。）。每一个模型的**核心结果也一定要说**，否则评委不清楚你们是否完成了题目要求（至少表面上如此）。
> - 到此就可以结束了，是否有必要写总结或者其他内容，团队自己决定，这属于非必要内容，取决于具体小题。如果倒数第二个小题（倒数第一题往往都是写信/写memo什么的）是用团队模型跑更多的数据，那这时候就应该写一段有关模型结果的。这样摘要才显得完整。

贴一下我们的摘要，可以看看是否和我的通用思路是对应上的。

Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2406176

Summary

Momentum, elucidating the phenomenon wherein a player's real-time performance surpasses pre-match expectations, holds paramount significance in tennis matches. This paper introduces the **Real-Time Momentum Quantification (RTMQ) model**, which quantifies momentum by assessing the disparity between a player's pre-match win probability and the real-time win probability. The study establishes the importance of momentum by comparing our model's results with an alternative model that neglects in-match performance. Notably, specialized indicators are employed to predict momentum swings, and game theory is applied to strategize against different opponents.

First, we use the independent and identically distributed (IID) assumption to reverse calculate the server's probability of winning the first point (p_0), base on the **Elo rating system**. IID assumption is not applied to calculate any other value, including real-time win probability. Then, **RTMQ model** is constructed by integrating p_0 and the **Real-Time Win Probability (RTWP) model** for calculating subsequent point win probabilities during a match, that also eliminates the effect of the serve side advantage. The resulting momentum difference aligns seamlessly with real-world observations. The **significance of momentum** is confirmed through a **K-S test**, rejecting the hypothesis that in-match performance is inconsequential. Further analysis involves applying the **Fourier Series Fitting (FSF) model** to track **momentum trend**.

To forecast large momentum shifts, a **Swing Prediction (SP) model** is introduced. The model operates on a per-set basis, as a set provides a more comprehensive dataset for analysis compared to individual games. We sample the momentum trend value (MMTTV) from the momentum trend at the end of each set. This is because the frequent in-set momentum shifts can render predictions based on real-time momentum data at the end of each set unreliable. At the same time, key technical performance factors are distilled from specialized indicators through **factor analysis**. We then employ a **Multivariate Linear Regression model** to predict the swings. Our model demonstrates an impressive prediction accuracy of 90.44% on our test problem, with specific factors such as 2nd_serve_direct_won and Net_approach_point_won identified as crucial contributors to player 1's momentum.

The paper extends its analysis to dynamic strategic choices made by tennis players during matches, employing the **Mixed-Strategy Nash Equilibrium model**. **K-means clustering** is utilized to determine strategic clusters. The Nash equilibrium is calculated, offering insights into **optimal strategy combinations** against different opponents.

The efficacy of the proposed models is validated through testing on notable tennis matches, including the extraordinarily tight 2023 Wimbledon Gentlemen's Final between Carlos Alcaraz and Novak Djokovic, and the one-sided 2017 French Open match between Andy Murray and Andrey Kuznetsov. **The SP model exhibits high accuracy (85.32% for the Alcaraz vs. Djokovic match, and 88.34% for the Murray vs. Kuznetsov match), while the FSF model accurately captures momentum trends, taking into account the score situation and historical commentary.** Eventually, we pinpoint crucial factors for inclusion in future models.

Keywords: tennis; momentum; real-time win probability; Elo rating system; mixed-strategy Nash equilibrium; quantification model

4.4 速读论文

检索论文，首先确保关键词精炼准确。其次，如果使用的是**Google scholar**，按相关度排序，可以先通过**标题和内容预览**（也就是大部分字都是灰色，但关键词被突出强调，且行内有省略号的两行文段）判断是否是团队想要的，然后点开看就完事。尽量有直接读英文原文的能力，如果没有，请检索好用的英文翻译插件，我比较推荐沙拉查词（这个还是瑞祥哥推荐给我的），但我也很久没用过了，不知道还好不好使。

4.5 解决编程手和建模手偶发的沟通困难

论文手要做那个非常清醒的人，而且写论文本身就需要较强的逻辑能力。所以如果建模手和编程手的沟通过程中出现了不同频道或者是单纯遇到了一个难以解决的问题，论文手都应该及时下场了解情况，然后推进任务。

4.6 快速、美观绘图

毕竟论文手写论文就挺花功夫的了，这时候不应该让画图耽误论文手的进度。每个图最多画一个小时左右就可以了，不需要太精雕细琢。

4.7 手要快

主要指写论文的速度要尽量快，不要到最后让建模手帮论文手大篇幅的写论文。一些必须有但不重要的内容可以分出去，但尽量还是自己写，而且最好做到建模手和编程手出结果之前就写好了建模部分，等结果一出来就可以完成结果讨论。我自己是直接用英文写，不需要写的多漂亮，表达清楚意思即可，写好了之后丢给gpt或者newbing出一个参考，觉得不错就直接替换我的底稿。每个人习惯不同，最终的结果好就可以。

5. 一位优秀建模手的自白

5.0 前言

说实话，出了成绩到现在我都有点不真实的感觉，没有想到真的能拿O奖！对此还是有很多感触和心得想要用文字的方式记录，感谢楷钧哥这条经验帖让我也有机会系统地整理思绪（不然我可能就发条朋友圈结束了哈哈），希望可以以这种形式帮到有需要的uu们。
考虑到楷钧哥的分享已经相当完善了，我的这部分分享主要会以建模手的角度展开，以及因为其实我数模相较起来也算是才入门，用个人经历做一些小白向的补充分享。

5.1 对数模比赛的理解

我第一次知道数学建模还是在初三，看到一直关注的毕导分享的一篇国赛A题的建模思路，现在还记得内容是关于隔热服的隔热效果讨论，其中里面一条很有意思的假设是把人体作为圆柱形简化计算，第一次具象的理解了“物理学家的球体鸡”在解决现实问题里的体现，虽然全文充满了毕导式的幽默，算不上严肃，但是却启蒙我对数学建模，解决现实问题，产生了原始的好奇和憧憬，也在未来大学生活愿望清单里面记下“参加数模比赛”这一项。



然而转眼高中三年，高考出分大学录取，经过周围真正天赋型选手peer pressure的洗礼和高考的毒打，我对自己的数学天分有了相当清晰的认知，决定放弃未来参加所有跟数学物理相关的学科竞赛。在这种心理下，参加第一次国赛完全是机缘巧合。当时大一结束的暑假我8月初就提前返校打算推一推大创项目，恰好舍友烺煊为了和队友准备国赛也提前返校，聊天中得知过几天学校会组织数学建模培训，感兴趣的同学都可以去听讲。

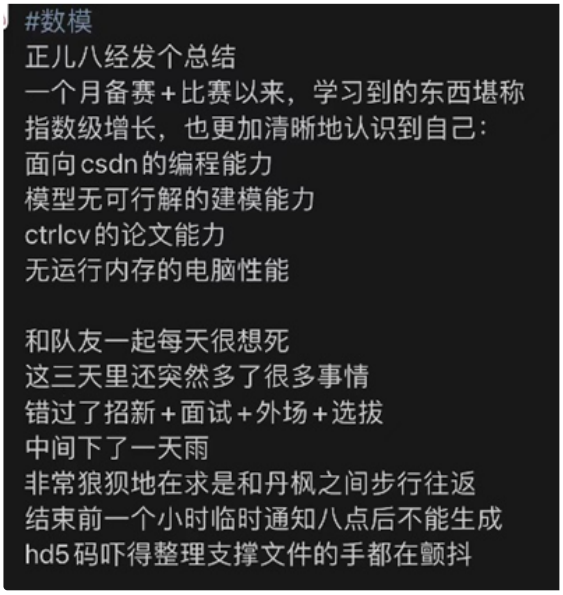
初闻没有放心上，一天偶然醒来突然回想起五年前读到的那篇文章，想起曾经的憧憬和想象，惊觉自己已经因为长久以来在应试上的受挫忘记了最初对数学建模的兴趣，忘记了想要用数学模型解释世界，理解世界，解决现实问题的初心。于是我犹豫了一会，想着反正不用额外交学费也没什么事，去听听课多学一点也是收获，正好培训就在第二天开始，我一节没落下的跟完了培训。

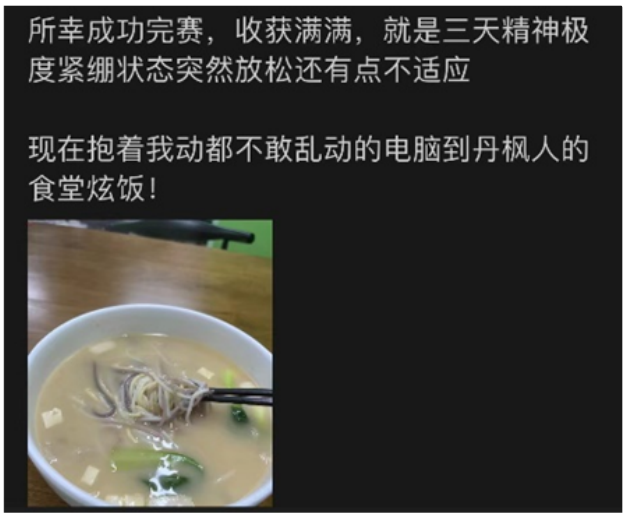
到达培训教室之后遇到很多认识的同学，老师也普及了很多比赛细节和注意事项，身处在一个大家都在讨论数模的学习环境里，我突然也拥有了“我也可以试试看”的想法，当晚就开始私聊同学、发小坦克帖子找队友，最后赶紧赶忙，在离比赛开始不到一个月的时候成功拉好队友，联系老师报名参赛。

第一次参加国赛可以说相当没有经验，当时对自己的定位和分工也没有很清晰的认识，又是临时找的队友，三个人的能力其实相当重合，幸运的是当时遇到的两位队友都是非常投入和认真。尽管中间遇到很多问题，其中一个队友比赛期间还赶上小学期作业ddl，但至少大家都倾尽所有，最终获得了相当积极的参赛体验，三个人都是第一次国赛还好歹拿了一个省二。

经过这次国赛的参赛经历，体验到自己从头到尾完成一篇论文的成就感（虽然还是比较拉），我真真切切地感受到自己拥有用数学模型解决现实问题的能力，认识到自己存在的问题和未来的改进方向。有种“一脚迈进数学建模大门”的豁然开朗感，再也不是只能存在于想象中的遥不可及的憧憬。于是下定决定要参加下一次美赛，吸取国赛的教训取得更好的成绩。

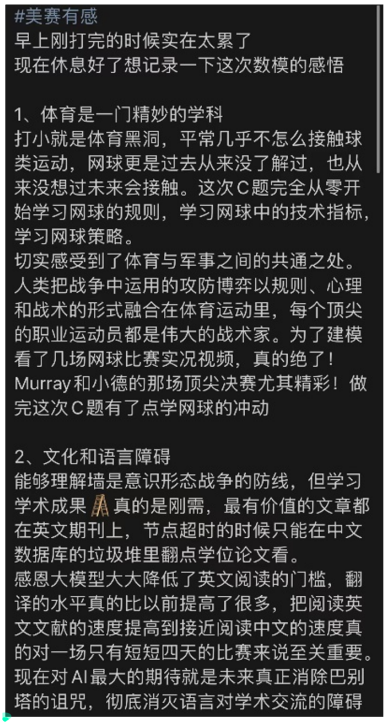
Po一个当时国赛结束的朋友圈：

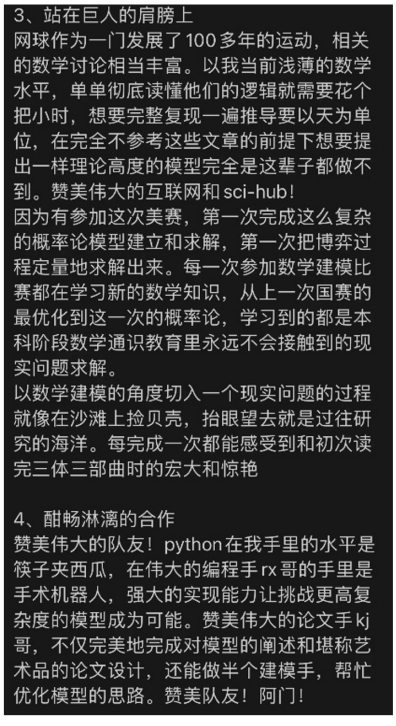


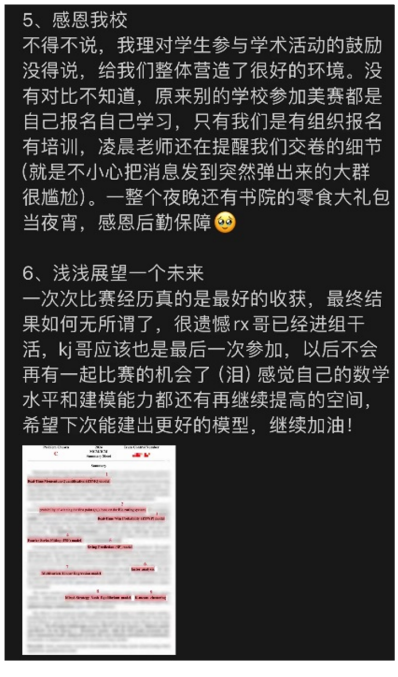


第二次参加就是这次美赛，吸取了上次和队友能力重合度高的教训，我早早就开始关注坦克上的组队贴，明确自己擅长的方向是建模为主，编程和论文为辅，也去和周围有数学建模经验的朋友交流。很幸运最后和楷钧哥和瑞祥哥组队，三人分工各司所能非常完美（这个组队过程楷钧哥写的很详细了我就不额外描述了哈哈）。

Po一个美赛总结的朋友圈，感觉已经把当时比赛收获写的很完整了：







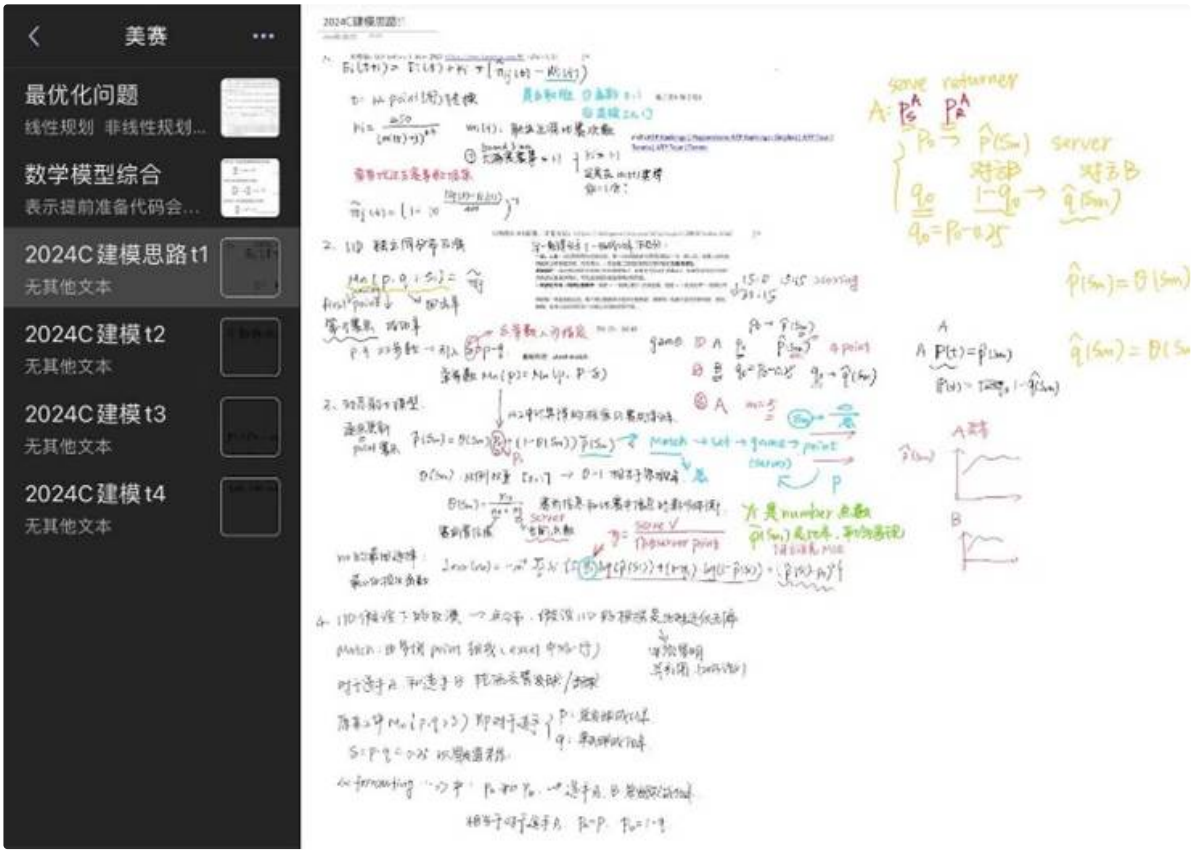
总结以上，其实这些经历于我个人而言凝练出的数模比赛理解主要是这样几点：

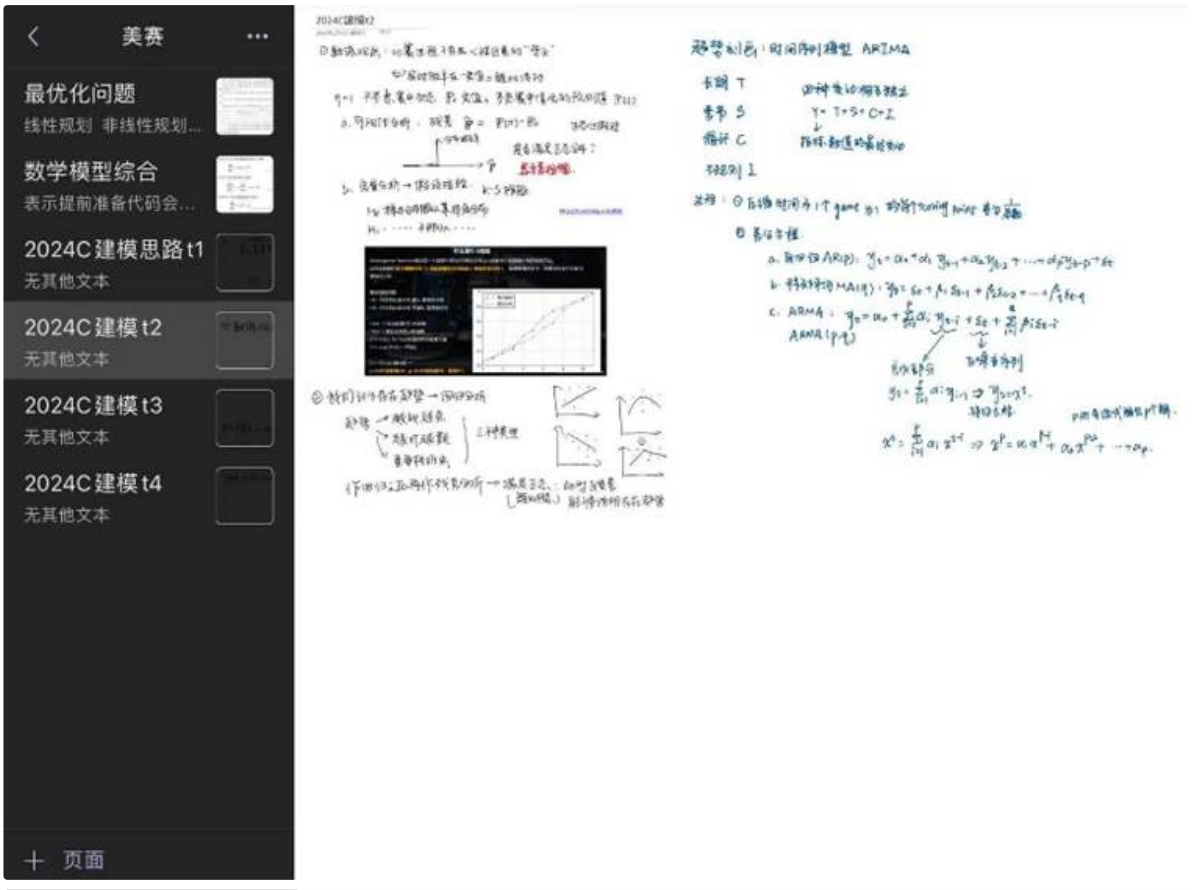
1、 数学建模其实有别于数学，其重心在于建模，这是一种与做数学题完全不同的体验，更准确的说更符合解决工程问题的智慧。我觉得对于所有对解决系统工程问题感兴趣的同学都很值得参加数学建模比赛，不必因担心自己的数学能力而失去尝试的勇气。很多能力就是在比赛中锻炼的，不是只有准备到100%才可以上，而且数学建模里的数学更重视将现实问题抽象化的能力，提高自己的认知的广度比单纯从深度上挖掘更重要。
2、 数学建模找到合适的队友非常重要，重点在于能力要每个人有各自的特长（这个kj哥讲的也很好我也不过多描述）。但是如果是新手小白第一次组队，其实很多人往往还没有想好自己的定位，我觉得这个时候最重要的是和真诚和认真的队友组队，大家能够全情投入，打完全程，就已经是最好的，而且有的时候跨专业组队也优势，在遇到同一个问题时可能会有不一样的见解。
3、 没有准备好也没关系，比其他人开始得晚也没有关系。重要的是对自己有所规划，以及和队友要有所规划，对学习新的知识始终保持开放的态度，但是一定要确保自己有足够扎实的基础，这一部分我放在了下一小节。

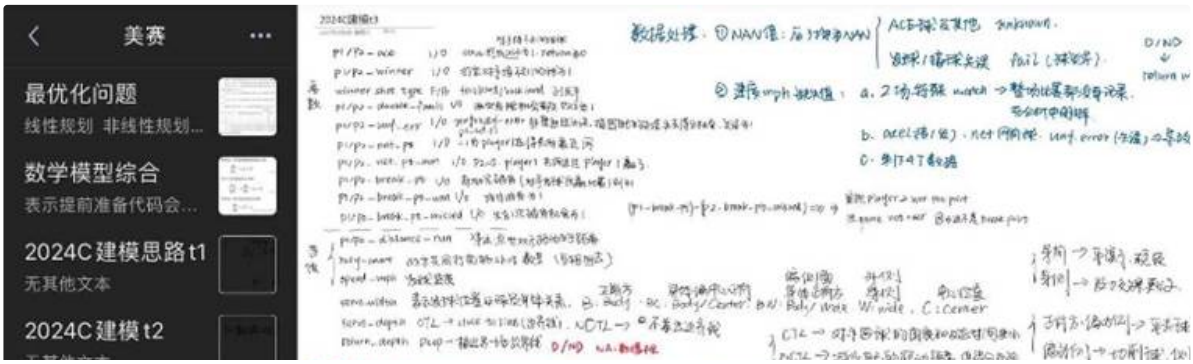
5.2 成为建模手需要什么

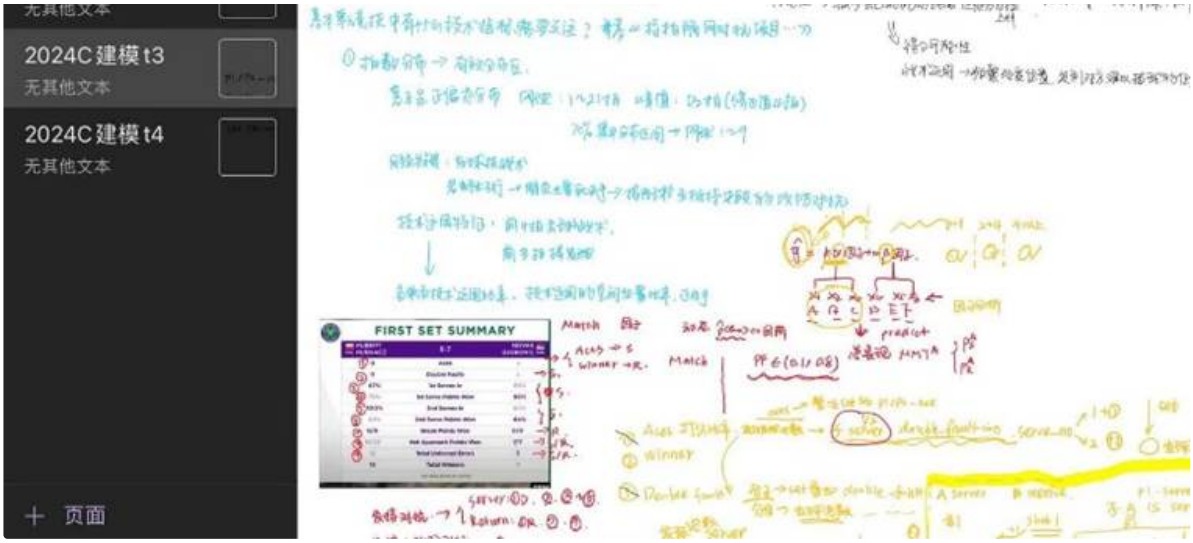
这一节主要是一些建模手的干货。

首先我认为，打数模，绝对不能只有建模手一个人懂模型，建模手必须要能研究搭建模型，编程手和论文手同时也一定要对模型有足够的了解，同时比赛中三人之间的高效沟通是非常重要的。这次美赛的模型，基本方向首先就是比赛前期我们三个人共同确定的，而在比赛进入中后期之后，具体的模型实现则是我和编程手瑞祥哥大量交流确定细节，而我再去繁取精把模型用于展示的部分传达给论文手楷钧哥。在模型的实现上，我们主要采用OneNote交流，因为方便我在iPad上手写公式，电脑上同步更新，随时投送给队友，附一些当时的记录：


美赛
最优化问题
线性规划 非线性规划...
数学模型综合
表示提前准备代码会...
2024C建模思路t1
无其他文本
2024C建模 t2
无其他文本
2024C建模 t3
无其他文本
2024C建模 t4
无其他文本


美赛
最优化问题
线性规划 非线性规划...
数学模型综合
表示提前准备代码会...
2024C建模思路t1
无其他文本
2024C建模 t2
无其他文本
2024C建模 t3
无其他文本
2024C建模 t4
无其他文本
+ 页面


美赛
最优化问题
线性规划 非线性规划...
数学模型综合
表示提前准备代码会...
2024C建模思路t1
无其他文本
2024C建模 t2



我觉得手写是更适合思考的记录方式，不过在比赛中，因为手写相较更难找到重点的部分，在编程手实现公式的时候很容易漏掉变量或者误解变量，所以比赛四天我们基本从起床就连着屏幕共享，随时交流进度，以及每过一段代码都要检查一遍。

同时，在具体的比赛过程中，模型中很多中间公式，往往一层嵌套一层，对于建模手来说很好理解，可以多个公式表达，在纸上写出来非常简洁明了，但在实现中，建模手需要转换编程思维，这个时候一些中间变量的定义对节约编程手时间和排查问题有很重要的作用。这一点非常需要编程手和建模手之间的有效沟通，我印象非常深刻的是在比赛中，因为涉及到的变量非常多，光在定义上我们就进行了大量的讨论，中间也伴随了很多建模手被编程手说服修改模型的过程。

除此之外，在代码跑出结果，前几版往往都是存在问题的，我认为排查问题的这个过程是体现数学建模工程性的很主要的一点。数学建模里所解决的问题来自现实，往往数据上就存在很多需要剔除的问题，在模型中往往也需要考虑数据现实特征带来的变量自相关性对模型产生的影响，有的时候也许是模型对变量的定义和现实不够符合。所以在具体实操中，往往是一边看着更改数据处理方法后得到的结果产生的趋势，再一边不断调整处理方法，从一次次试验中思考总结不同变量的影响，才能最终得到有意义的结果。而在备赛时学习的数据处理方法所能处理的问题往往都很简单，这种能力是非常需要从实践中锻炼的。

以上所提及的主要是建模手与队友的合作需要注意的要点，沟通真的是非常重要的一环，在比赛中想要实现更好的模型效果一定是具有大量讨论的。这里也必须要说，专业的编程手对于模型建立真的有非常重要的作用，在国赛时我和队友都是非计算机专业的学生，由于编程实现能力不够，实现速度不够，前两题还算较好地建立并求解了模型，但实在没有时间和精力继续以同样的复杂度完成最后一题，我认为这是当时国赛失利的主要原因。

接下来是建模本身。我认为每个建模手都必须在赛前建立好自己的模型知识库，了解基本模型是绝对必须的。但数学建模本身是个涵盖面非常广的学科，在具体的备赛上对模型的学习一定要有所侧重。比如我国赛时作为初学者，早在备赛时就决定好B题或C题作为比赛方向，所以整个暑假的备赛过程我都专注在最优化问题和评价模型，而美赛时则是主要针对数理统计和概率论这一块。学习模型主要就是阅读教材+b站大学自学，网上有很多免费的资源都可以获取。附一个当时规划问题的模型计划，说实话，大部分内容我看完都不记得了，但是这些表可以帮助我迅速找到需要的内容。

★ 目标规划

目标变量、约束条件、偏差变量、优先权

优先权目标规划：建立单一目标线性规划--->求得最优值作为约束--->考虑下个目标（较低优先权不改变较高优先权）

加权目标规划：量化优先权，分配罚数比重

★ 动态规划

将多阶段划分为单阶段

★ 图论

点、边、弧（有向）

无/有向图、连通图

最短路问题——Dijkstra算法（双标号法）

最小生成树问题——树：无圈连通图，破圈算法

最大流问题——线性规划、网络图论解法、最小费用最大流

★ 存储论

确定性/随机性存储模型

是否允许缺货，数量与单价折扣

需求之间存在依赖关系——物料需求计划（MRP）

★ 排队论

（1）组成部分

通道数——单通道/多通道

顾客到达——泊松分布

服务时间——负指数概率分布

排队规则——损失制/等待制，先到先服务

平稳状态/过渡时期

（2）模型实例

M（泊松）/M（负指数）/1（单通道）/inf（排队长度无限制）/inf（顾客来源无限制）

M/M/c（多通道）/inf/inf——增加经济分析（成本）

M/G（任意分布：均值+均方差）/1/inf/inf

M/D（服务时间为常量）/1/inf/inf

M/G/c/c（排队长度有限制）/inf

M/M/1/inf/m（顾客源有限制）

M/M/1/K（系统最大容量）/inf

M/M/c/K/inf

（3）应用

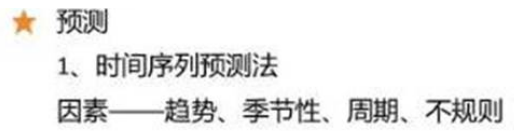
生灭过程——M/M/C or M/M/C/R 平衡：流入=流出

★ 决策分析

1、不确定情况：最大最小准则（悲观）、最大最大（乐观）、等可能性（期望）、乐观系数准则（乐观悲观折中）、后悔值准则（沙万奇）

2、风险型：最大可能（转化为确定）、期待值准则、决策树（多级规划）、灵敏度分析、EVPI（全情报价值）、具有样本情报、效用理论（盈利损失转化为效用值）

3、层次分析法.etc



这种作为主要方向的模型的掌握程度需要达到：
1、 能够理解这个模型的推导过程，选择这样做而不那样做的意义
2、 这种方法存在的缺陷，当数据具有什么特征时适用而什么时候不适用，当不适用时又可以转而选取什么方法。
3、 模型具有什么变体，结合什么方法可以优化已有的模型。

除此之外，还需要补充对往年题的练习，建模手一定要广读往年题，先读题，自己思考，再去看优秀论文。要总结归纳，现实情况出现什么时，适合选择这个模型，自己的思路和优秀论文的思路有哪些差异，优劣之分在哪里，这些都可以补充进你的模型知识库里，成为你的个人见解。

而在广读往年作品时，往往会了解到一些超出你所知道的基础模型的方法，这些额外的，在教科书外的东西都可以拓展了解，这些当时只是了解一下的内容在比赛中遇到具体问题时都有可能成为你“乍现”的灵感，为问题的解决增彩。

比如这次美赛中，看到题目我们一开始遵循着往年题目的惯性，第一时间想到的都是概率论的做法，但反复阅读题目后，我发现，普通运动员比赛寻求胜负规律确实可以以概率的方式描述，但顶尖运动员网球运动，在数学本质上，更贴近的叙述应该是博弈论而非概率论，因为顶尖运动员之间的水平差很小，胜负往往就在一线之间，而在我们所能获得比赛数据中，除了胜负场次本身还能呈现各种比赛策略的选择，更适用于博弈论。这个灵感就是出于赛前读到的一篇提及博弈论的往年论文，我其实毫无知识基础，比赛中现学现卖，但最后问题解决实现的效果还是很好的。

最后，我认为建模手一定要勤于思考和钻研，现实是复杂的，精准的模型一定也是复杂的。第一次国赛前我和队友也做过一次赛前模拟，当时建立的模型是比较烂的，为了简化实现难度自作主张的使用一些线性的关系，这种是非常不可取的。所以在国赛的时候我有意改进自己的模型，牺牲复杂度和实现难度，也要达成较好的精准度，虽然为此后期非常赶，而且实际上没有很好地把握进度。这次美赛也是多亏楷钧哥非常有参赛经验，一直牢牢把握整体完成度。但我认为，追求更精准、复杂度更高的模型应该作为每一个建模手的理想，具有这种理想才能支撑对问题的不断深入思考和对更多知识的学习，以及对数学过程的推导。（当然，作为队长就还是要锻炼自己把握项目进度的能力！）

5.3 心得和展望

以上是一些我个人的心得，但说实话，我认为自己作为建模手还远远算不上顶尖选手，这次能够获奖很大程度是因为有幸组成了非常完美的队伍。作为一项3人组队的竞赛，在这次比赛里面，我们三个人都确保自己的部分圆满完成，同时有效沟通互相促进，让1+1+1>3，我想这才是我们这次获奖的决定因素。

数学建模也让我重新思考自己的专业，认识到“工程思维”和“工程能力”在现实问题中的体现，也让我认识到自己，对自己的专业确实充满兴趣，希望未来能够继续在各种比赛中持续锻炼自己多方面的专业能力。这种自我探索的过程我认为也算得上参加各类比赛的一种意义，非常鼓励有兴趣的朋友尝试。我想，这种自我成长的过程，才是绩点和优绩主义以外，真正的大学生活啊。

6. 论一个称职的队长

基本上大多数队伍都还是有一个队长身份的，个人认为称职的队长主要应该在以下方面做的比较好。

6.0 对队员的绝对尊重和耐心

数学建模比赛里，队长比队员多了引导和做决策的任务，但所有决策必须是队友全部认可且同意后才能执行的。也就是说，相比于拥有“你俩全得听我的！”的权力，队长更多是担负着“引导思路、对团队的不同选择进行权衡利弊以便队友做出判断”责任与义务，很像是“队长”和“分析师”的兼任。所以，必须对队员有足够的尊重和耐心，每一个决策中三个人的权重应该是几乎一致的，当然有经验有能力者的话语权自然更大。当然，这一点的前题是队友都很愿意配合（有理有据的质疑和反对不属于不配合）&时刻保持理性在线。

6.1 果断——把控团队解题进度

队长必须能够对团队的解题进度有良好的把控，不至于安排过紧，更不至于前期过于松垮后期紧锣密鼓导致忙中出乱。这需要比赛经验，也需要临场判断。团队最经常面临的一个岔路口是，死磕某个小题或小题中的某个关键问题很久后，是否继续进行建模或者求解。我们经常采取的措施是立ddl，即到ddl了还没做出来，就发挥用最快的速度发挥创新思维解决问题（创新思维就不细说了），取一个合理的参考值继续往后做。如果最后有时间再回头磕遗留问题（一般都没时间）。涉及到任务推进，只要到了ddl，都需要队长主动提出这件事情，保证团队最终可以完成所有模型的建立和求解，并且写完论文。

6.2 保持冷静

每次数学建模比赛基本都会遇到至少两次以上的大坎，此时需要队长尽全力保持冷静，然后分析问题的关键，进而带领队友一起找相关解决办法，保证先能够做出结果，不必追求解决方案的完美。

7. 如何找队友?

7.0 理想的论文手/建模手/编程手的模样

首先，论文手和建模手可以是理工科、商科，因为需要保证其逻辑能力和数理基础。论文手写作能力可以从**课程论文/个人简历**中确认，并且**优先选择懂建模、英语好 (四六级、雅思托福gre) 的论文手**；建模手的建模能力应该从其**对各类模型的理解和认识**中确认。编程手最好是**计科**学生或**长期有编程习惯**的学生。

简单介绍一下我们团队吧顺便。我，论文手，论文能力强，英语好（雅思8），了解常用模型及其分类，文献检索、数据调研能力较强，美工能力尚可。csq，建模手，国赛北京省二（当时她是编程手），建模经验几乎0，但寒假期间认真备赛，了解了各种模型。瑞祥哥，打过ACM，计科研0，研究方向是多模态，擅长数据题C题，以及图论D题。

下面分陌生人组队和熟人组队两种情况讨论如何找队友，以及找队友过程中的一些有助于让团队更可能走向成功的细节。当然，我坚定地认为三个人里有一个有经验的即可，另外两个人就算是小白，也完全可以来从零开始备赛、参赛，甚至有希望第一次就拿到M及以上的奖项。个人认为，“可合作性” ≈ 个人能力 > 参赛经验。

7.1 陌生人组队 ⭐ 😈 / 👼

1. 确定对方的经验，让对方发简历、相关比赛证明资料
2. 组队前尽量线下见面（实在不行线上也可），随便聊聊，过程中问如下关键问题：
    - 为什么想打数学建模比赛?
    - 是否清楚数学建模比赛**需要脱产进行**?
    - 比赛的几天是否有其他更重要的事情?
    - 最有效的预防“卖家秀买家秀”（ta自己说的和ta实际做的高度不一致）的办法是，组队的三个人一起签字立君子协定：除非有重大事项发生，否则不可以做出无故玩失踪、不配合参赛等行为，如若发生，将所发生的事情进行客观陈述、处理隐私（成员自身人名公开时不做处理）后公布到校园平台、校园数学建模比赛群。
3. 在备赛阶段的**早期**至少进行一次为期至少一天的模拟赛。

经过这三个步骤，如果还是遇到了不靠谱的队友...那只能说是天意，你会在其他事情上好运的，嗯嗯。

7.2 熟人组队

最好是选择这个方式，因为彼此之间较为熟悉。优先选择善于团队合作、能够接纳包容不同观点的身边同学，朋友也可以。但是要进行正式的沟通，明确大家打比赛的意愿，要一起努力备赛。同样的，最好在备赛早期进行模拟赛，确定大家的水平和对待比赛的认真程度。

8. 如何进行备赛最有效?

8.0 对于整个团队：周会 & 模拟赛⭐

不论团队中每个人的水平如何，我都认为开周会和模拟赛是最有助于团队整体能力提升的两个方法。

- **周会**：每周成员们聚在一起，每个人用1-5分钟讲上周都做了什么备赛工作、自己有哪些收获（知识/技能）、下周自己的安排等。然后大家再花一点时间进行相互交流，互相提建议，有经验者要多提具体的建议以供队友参考。我给自己的安排经常是一周读1-2篇O奖论文（我大三寒假也挺忙的），然后过程中多复现漂亮图（核心模型流程图就是我受到启发后自己画的），然后每周进行一次个人模拟赛，练习**分析题目、研究问题、找基础模型**的能力（这个是进阶内容，但我觉得这对论文手真的**非常非常有帮助**）。
- **模拟赛**：举行一两次足够，每次最少4小时、最长100h（即全程模拟，不过我觉得没几个队伍能做到，并且也没啥必要）。**团队三个人一定要在备赛开始之前确定一个题目用来打模拟赛，这样各自备赛的环节就不要看这个题目的o奖论文了，否则就起不到模拟效果了。**目的是在实践中磨合队伍，对团队研究问题、定基础模型、论文手的逻辑、建模手问题转化能力（建模能力）、编程手实现能力等能力进行检验。是个很好的小试牛刀的方法，赞。

下面具体来说论文手、建模手、编程手三个人各自应该完成哪些方面的备赛。

8.1 论文手⭐

- 读呀读呀读论文！O奖论文最少看他个三五八篇的。然后，第一次参赛的论文手必须阅读《正确写作美国大学生数学建模竞赛论文》。不需要做特别详细的笔记，粗读的过程中高亮那些你觉得对自己真正写论文特别有帮助的内容即可，也可以写一点备注记录自己的理解，方便比赛时快速复习。
- **个人模拟赛！**可以选定以往美赛的某道题，然后自己模拟一遍参赛过程：读题，分析每个小题的要点和破题关键，搜论文，确定基础模型。也就是把论文手应该做的最利于团队的事情进行演练和熟悉。其实，我觉得个人模拟赛是非常能够考验论文手能力的手段，能够**非常高效地磨练论文手，使论文手快速适应数学建模比赛的节奏和套路。**
- 用ppt等软件复现各种各样好看的图！为了比赛中能够**又快又好**地画图。当然，每种图储备3种左右即可，**储备胜在类型丰富，而非形式多样但类型单一。**
- 了解建模！b站有大量建模相关的内容，如清风、数学建模老哥等up对数学建模的入门级教程。请所有论文手懂建模，除非你想在参赛时瑟瑟发抖、听建模手和编程手的讨论仿佛在听天书。

  我个人并没有系统学习过数学建模，我在大一的暑假出于兴趣入门了机器学习（coursera上吴恩达的课程+b站的黑马编程师三天快速入门机器学习，个人学习笔记见python机器学习快速入门_pyhton的训练模型入门-CSDN博客）并浏览了深度学习相关内容，以上就是个人几乎全部的基础（当然，评价、预测等经典模型也大致了解）。其实不需要论文手（甚至建模手也是）多会建模，许多O奖论文的模型非常朴素，如2024年C题中，2401445团队的论文（O奖+$10,000奖金），核心模型是一个朴实无华的多元线性回归模型。。。非常佩服。
- 建议论文手使用latex写论文，因为格式编辑方便快捷。如果**实在学不会，再考虑word**。如果论文手无法自己独立学会使用latex写论文，那么本次比赛大概率是和F/O无缘的（不使用和学不会是两个概念，学会latex仅需要较低级别的问题解决能力）。我们这次比赛是用的**overleaf**的2024美赛模板写的，在线编辑非常好使，也没崩过。使用texstudio等软件也可以，都可以，没啥大差别，毕竟主要负责写论文的是论文手，建模手和编程手进行在线编辑的频次并不多，但好处是建模手和编程手可以实时看到论文写作进程，**对于经验比较少的朋友们还是建议使用overleaf在线编辑。**

  当然，我已经把我们论文的latex源文件只读链接放到了文档里，所以不会使用latex的问题应该不会存在了，有些细节自己懒得学的话及时用newbing问一下即可，我就经常这么干。再放一下🔗：https://www.overleaf.com/read/fpjcfrjrjnrd#cd1c9b

8.2 建模手

- 懂基础模型！美赛按照赛题，分为A题连续型（微分方程）、B离散型、C大数据、D图论（E和F无法进行类似的总结，不过经常用到评价模型、预测模型等）。至少要会的包括微分方程模型、优化模型、规划模型、分类模型、聚类模型、评价模型、预测模型等等模型，具体的模型清单到处都有。一定要熟悉模型 的概念和应用场景，最好能够了解模型求解相关知识和案例（方便和编程手进行沟通），不需要掌握模型的推导过程。

- 读O奖论文的建模流程！尽全力对建模流程进行理解，加强自己建模的专注力。建模手经常会遇到需要自己进行细致手算甚至是推导模型关键环节（联系模型和具体问题，推参数的计算公式，等等等等）的时候，而且一干就是两三个小时。这是建模手必备的能力。

8.3 编程手

- 建议人人学会使用github copilot辅助编程，在生成代码的准确度上比gpt和newbing好太多。如果编程手是计科学生，只需要让编程手尝试复现那些在附录里贴了代码的O奖论文即可。
- 编程手自己一定要清楚自己擅长的编程领域（覆盖A~D中的两到三个题目即可），然后备赛期间加强对相关领域模型的算法实现的知识。一般python就够用了，最多加个matlab搞优化，微分方程那些。
- 更多有关编程手的备赛我就不了解了，总之如果找到一个计课的编程手，ta的备赛基本就是去复现附录里有代码的O奖论文，来练练手。尽可能做到**看论文（不看附录里的代码）的模型建立部分即可成功复现结果**，如果做到了，那就绝对是一个优秀的编程手了👍。

9. 比赛前，可以为比赛准备什么?

9.1 论文手

- 论文的latex模板（包含三线表、伪代码、等等等等）

  直接看我们的overleaf在线编辑项目即可，我可以说我前四次比赛都没有一个准确&好看的伪代码模板，这次一篇o奖论文所需要的全部内容模板全在https://www.overleaf.com/read/fpjcfrjrjnrd#cd1c9b里了。
- 漂亮图
- 往年o奖论文（对于A~F六道题，把每道题其中某一年得了informs奖或者comap scholarship的o奖论文保存下来，得到6篇标杆o奖论文），便于比赛时打开，照葫芦画瓢

  我有个习惯就是不论做什么题，都打开一个往年的o奖论文。今年是我第一次写C题的论文，我就打开一篇c题o奖论文，模仿它的大纲和建模过程。照葫芦画瓢错不了。
- 使用newbing搜数据的能力

  多用用newbing搜东西，写prompt熟练一点。2024了，论文手应该熟悉如何编写提示词才能让大语言模型的生成更准确、更具体。这样团队的检索环节就会大大提速，准确率也大大提升。

9.2 建模手

- 常见数据库
- 常见模型的介绍链接，通常在csdn上

  到时候要用哪个模型直接把相应模型的🔗发给论文手，让论文手有个参考。

9.3 编程手

- github copilot
- 你觉得特别有助于美赛模型实现的代码
- 常见模型的代码（我分享的百度网盘链接里有网上流传比较广的【但可能没啥大用 哈哈】。网上都有，但是基本上用不到，不过这里还是写一下，给这些非常传统的一点也不具体的作用极其有限的模型一点尊重）

10. 比赛过程中我们使用到的工具

我浏览器过去三年里收藏过的美赛相关的网页都在这里，有不少是没用到过的，大家看看就行：

> 1. MCM: The Mathematical Contest in Modeling

2. https://pan.baidu.com/s/1n7RxM-l_n85g19TyHz7sRA?pwd=2024

   2024年美赛O奖论文合集。此外，前些年的，网上也都有。2024年C题的**2401445**队伍（O奖 + *COMAP Scholarship Award* ）的论文我觉得很不错，使用基础模型得到了很不错的结果，可以多多学习。

3. (14条消息) 一份简短又全面的数学建模技能图谱：常用模型&算法总结 *冷月无声的博客-CSDN博客* 数学建模模型
4. 数学建模算法汇总 | GeminiPlanet
5. (13条消息) 经典中的经典算法:动态规划(详细解释,从入门到实践,逐步讲解) *ailaojie的博客-CSDN博客* 动态规划算法
6. 镝数图表 - 在线数据可视化神器，让你的数据有型有款、创意无限
7. MATLAB 2020a安装教程和破解方法(附密钥) | 我爱分享网
8. SPSSPRO-免费专业的在线数据分析平台⭐

   这个确实好用，很多模型都有，给数据后可以直接跑。不熟悉建模编程的朋友们欢迎使用。

9. 数学建模美赛写作模版(包含摘要、格式、总结、表格、公式、图表、假设) - 豆丁网
10. 美赛获奖指南_编程
11. (13条消息) MatlabR2021b自学笔记一：安装教程（附安装包链接）_哆啦lalala的博客-CSDN博客
12. Create LaTeX tables online – TablesGenerator.com
13. 10类案例带你了解论文插图制作 | Levitate_
14. Apache ECharts
15. Graph Editor（不知道干什么的）
16. 美赛最全软件汇总及使用说明
17. 数据提取神器GetData使用教程 - 知乎
18. getdata软件_【安装教程】图形数字化处理软件之 Getdata v2.24_weixin_39519554的博客-CSDN博客
19. 时间序列分析 | 相似性度量基本方法 - 林胜联府 - 博客园
20. LaTeX Template for MCM-ICM - Online LaTeX Editor Overleaf

    overleaf里搜**LaTeX Template for MCM-ICM**即可。这个链接是我搜到的2024美赛模板。

21. 在线LaTeX公式编辑器-编辑器

10.0 团队论文调研平台

- Google scholar

  找核心模型（MMT量化、实时胜率计算）

- 知网

  找后面几个小问的模型（博弈论相关）

10.1 团队沟通平台

- **钉钉**会议

  多人不限时长会议

- **OneDrive**共享文件夹

  任何需要大家一起看的资料都会保存到这里，包括参考论文、所有结果图、数据集、 代码等。

- **Microsoft OneNote Online**

  可以用来管理团队进度（建议新手利用这个方法，否则容易遗忘发布过的任务），但主要是用于团队成员间沟通。建模手可以将详细建模流程较为工整地写在这里。ipad和电脑都可以访问，因此可手写可打字。

10.2 论文手使用到的

- Overleaf——在线编辑LaTeX

  写论文的，好用，不用担心论文因未保存而意外丢失。
- PPT

  画图（当然，鼓励使用其他更好的方法）
- matepad自带的笔记软件

  写思路、列to do list等

10.3 建模手使用到的

- Microsoft OneNote Online

  用来写建模的详细流程；记录自己观点等

10.4 编程手使用到的

- Jupyter Notebook

  敲代码。
  - Github Copilot

    计科研究生们普遍使用的AI插件，提升编程效率

11. 推荐一些公众号

不是引流，我看到一些公众号给出了较为详细的建模流程+基于python的算法实现，包括但不限于如下公众号：

- 机器学习和人工智能AI
- (不再详细条列，因为好多公众号名字我都不记得，写上面那个单纯是刚刚看到一个不错的)

只要你经常浏览这些公众号的文章，在**订阅号消息**内你就会不时收到相关文章（不仅限于该公众号）的精准推送。当然，最好的办法是把你想要使用的模型名称在微信搜索框内搜索。比如搜XGBoost，搜随机森林，搜模拟退火。**微信搜索，真的很好用。**

11.1 一个好用的搜索小技巧⭐ ⭐ ⭐ 😲

> 如果你在微信搜索内搜模型，请搜索【编程语言】【模型名称】，如`python 模拟退火`。然后，如果搜索时没有带编程语言，**优先选择那些公众号名称里带python，其次是带matlab的文章看。**你会来感谢我的。

12. 美赛带给了我什么？ ⭐

其一，也是我认为最重要的，逻辑能力的强化。美赛相当于团队在100h内要针对一个专业领域的问题给出自己的详细解决方案，可以认为是一次时间紧张的超短期科研。美赛需要成员时刻保持逻辑思维在线，才能高强度的完成整个问题解决方案的设计。我能够体会到自己更容易抓住专业课里实验环节的核心流程，且平常写实验报告的过程中会感到思路更加清晰。

其二，科研写作能力，尤其是模仿能力。一切论文的写作源于模仿更优秀的论文，此外语言表达能力是通用的。

其三，调研文献的能力。如何根据问题确定搜论文的关键词、在读论文的过程中最快地筛选出对团队建模有帮助的文献、通过搜论文熟悉陌生概念等。

其四，团队合作能力。毕竟三个人需要保持高强度的分工合作、并且各司其职，才能确保团队整体效率较高。

此外，一个对升学或许有一些用处的奖状，以及和队友的友谊。因此，我还是十分推荐理工科/商科朋友们来参加数学建模竞赛（特别是美赛），这个比赛能够全方面地提升大学生的逻辑思维、科研写作、文献调研、团队合作能力，且门槛很低。同时，我并不认为参加数学建模比赛是一个能够非常有效锻炼建模能力/编程能力的赛事，很多博客也提到了这一点。**相比于提升学生的专业技能，我倾向于认为美赛更能够全方面提升学生做科研的基础能力。**

后记

我从初中开始就经常输出几千字甚至上万字的小作文 游记等等。文字承载着内心最本真的想法，真情流露并不会骗自己。每一次记录中，我视自己为客体，目光浏览过回忆里的一幕幕，写下尽可能多的思绪。若干年后的自己，当你阅读这些的文字，希望能够回想起你看似卡皮巴拉（你还会记得这个词的含义吗？）实则丰富多彩的四年青春。

后记是我最开始写的内容，因为三个月前（2024.2）的我无法想象自己会获得这样一个想都不敢想的奖项。2024年美赛C题全球（当然，大多数都是中国队伍）一共10184支队伍，O奖团队只有11个，仅0.1%的获奖率。数学建模从来不是我努力培养的能力，在我大学的时光里也仅是一条支线。未曾设想，三年前朋友的一句话，会让这条支线生根、发芽，在生活最不起眼的角落里成长，带来比主线任务更多的惊喜。

大一上，忽然有一天panda问我要不要一起打数学建模比赛，我一遍遍地听着陌生的词语，心底被激发出一点点兴趣，不知不觉的就开启了【副本：数学建模】。

无心插柳。

第一次参加美赛，和panda、dsq一起。我因为英语相对更好而担任论文手，panda编程，dsq建模。三个大一学生在panda的忽悠下，硬着头皮给熊cg老师发了一封邮件，说我们想打美赛，希望熊老师当我们的指导老师。第二天panda黑着脸(不清楚这个表达是否有艺术成分)跟我们说 熊老师并不建议三个啥都不懂的大一学生参赛，大概率100刀乐扔给组委会是听不到一点响的（事实证明确实如此，当然也有大一美赛直接拿F的）。但奈何我们够莽，吭哧就是上，选的还是MCM A题自行车竞速，论文纯靠忽悠，模型纯靠语文、草稿纸和铅笔，编程全靠excel人工计算，喜得H奖（top 25%）！开奖日我信心爆棚，对自己身为论文手的能力深信不疑。

第二次参加数学建模就是国赛，那是一次很不好的回忆。我没有参加校赛（大一下课外的重心放到了挑战杯和互联网+，不再和panda&dsq组队了，从此他俩走上了每打一两次比赛就换一次队友的路），不过当我真到了大二上，国赛快开赛的时候，我还是想打个国赛试试，于是就去找一个缺论文手的国赛队伍。在小坦克捞人，捞到了以为很厉害的两个大三学长，没想到人家开赛后根本就没把什么功夫花在国赛上，每天打比赛时常不超过3h，我几乎全权负责论文 建模 以及大部分的编程。那是最不愿回忆起的一次数学建模比赛，我做梦也想不到自己也会瞎眼碰到如此恶心人的队友。喜得成功参赛奖（陪跑，top 100%）。往事随风，这件事给我的一大教训就是重大事项的合作必须摸清合作对象的意愿。

第三次参加数学建模，又是一年美赛。微光灵雕项目组认识的学姐带着一位编程很厉害的计课编程手（没错就是瑞祥哥）找我一起打比赛，D题，结果分工出现问题，我在额外负责大部分论文的同时负责建模，模型全靠猜想、脱离实际，浪费了编程手的能力。结果也并不出意外，H奖。

第四次参加数学建模，又是一年国赛，我兜兜转转回到了panda和dsq的队伍。曾经弱小的18岁少年/少女，两年后已然成长为20岁少男/少女。其实我当时已经发现我和panda能力的重复度极高，不过还是由于合作一定会愉快的事实一起组队。赛前我信誓旦旦的说，最优化已经掌握到next level，有公式就能跑出结果（当然，本来给全公式就应该能跑出来），给我牛坏了。结果，一个所有参数（甚至变量的数量）都是变量的定日镜场，击碎了那自大的青涩编程手（我）的梦。在panda力挽狂澜下，硬生生把论文写的很漂亮，喜提省一！

此时我已经释然，隐约中明白了自己能力的极限。三次论文手，并没有好结果。一次划水的编程手，结果已经让人满意。我已经做好了金盆洗手的一切准备。

直到那天，直系学妹csq找到我，非常happy的告诉我她找到了一位编程很厉害的打ACM的计科学长，希望让我作为论文手加入队伍。

我说，这么帅的！那也不是不行，先沟通沟通，拉个群聊聊。

我点开群聊，看成员头像，“北理 20 计课 srx”映入眼帘，不禁一愣。

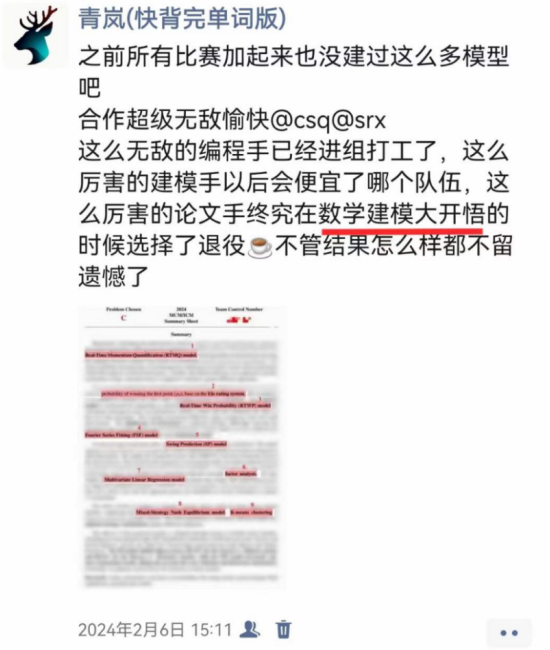
命运的齿轮从那一刻开始飞速运转。（你有这么高速运转的齿轮进入数学建模团队...）

我从未和csq共事过，但之前线上三言两语的交流、她高中的学校和生活、她平日的言论让我笃定这是一位非常值得与之合作的朋友。以及，第二次美赛和瑞祥哥的配合也很默契，并且我也非常信任他的编程能力。我就在想，那就再打一次吧。

这一次，想法很纯粹，不为奖项，只为写出一篇漂亮的论文。比赛前，我甚至有几次忙过头的时候都想过要放弃这次比赛。不过每次都觉得不应该辜负队友的信任，同时自己也并不需要大费周章进行备赛了，所以每次都凭着自己那份淡淡的兴趣和对团队合作的渴望坚定了选择。现在真是想谢谢当初坚持下来的自己，和鼓励我参赛的队友。

备赛，开赛，一切都很顺利。定C题，搜论文，找到符合下半部分模型的中文论文；继续搜论文，找到超级无敌符合上半部分模型（即模型主干）的英文论文；搜问题核心概念（momentum和swing）的定义，找到1985年古早论文的准确量化定义，联系起主干模型和概念，.....正如赛前的预想，一切合作都十分顺利，偶尔有思路出现卡顿的地方要么被我剑走偏锋甚至暴力推进（没错有的时候需要果决一点），要么被csq啃下来正面解决。截止时间前两小时，收工，发邮件提交论文，互相道别，然后即刻钻进被窝（毕竟通宵了）。

下午大家纷纷醒来，还是有点小兴奋，有点小快乐。我发了条pyq庆祝数模生涯的结束，csq总结了很多东西，但是她的pyq仅一个月可见，所以我现在看不到了。



三个人，六个大眼睛，就连我把摘要厚码发圈后，都没发现，我们的论文，没！！写！！标！！题！！！

这件事情还是我打完比赛十几天后和panda一起爬百望山的时候交换两支队伍的论文时才发现的。他说他们标题写的特别好，然后我当时就想我起英文名的功底也挺好啊，然后...然后突然发现自己似乎没有撰写2024美赛标题的记忆...瞪大眼睛一看自己的论文...

标题呢...不是哥们...我标题呢...

这也是为什么这个分享贴的标题最开始是《无题》（后来想想，还是写了一个标题，不然别人看着无题觉得无聊）...瑞思拜...

几天后我把这个坏消息告诉了csq，她真的好担心我们直接被判违规。当时提交论文的时候，也很担心我们超了规定页数2页会不会被判违规。我也有点慌，和同样打过美赛的高中好哥们（msh）聊了这件事，很快他发来了消息。他问我，“高考论文多少分？”“50啊。”“没写标题扣几分？”“2分...呃...很有道理。”然后我拿同样的话术安慰csq，最终我们都看开了。

我当时在csq朋友圈底下评论说，稳M争F，评委不给F就是不长眼。发完之后又觉得没什么底气，毕竟自己主导的三次比赛里，获奖情况是：h-成功参赛-h，所以历史KDA并没有给自己足够的自信。

没有想过开奖会是在劳动节，十二点了被喊起来起床吃饭，这才揉了揉眼睛拿起手机看上午的微信消息。一个个未读红点不断被清理，然后一个O不经意间路过我的眼睛，与之而来的还有panda的夺命连环call。

一声“啊？？”，五声“卧槽”，瞬间清醒。

参赛时的那份纯粹，就这样回报给了这支没有写标题的团队。

曾经的种子，如今的绿荫。

四年青春，朋友难得，同路人更难得，同频者未可遇。路途不远，但我很幸运，遇到你们。

学了三年工科，愈发感觉到，工程科学的本质是解决问题。

自我感觉并不是传统意义上的理工男，我并不对各种抽象的概念和一切涉及动手的事务有太大兴趣，但我却热衷于完成一个个小项目乃至研究，体验路上解决问题的过程。三年工科学习绝对不止带来了n多领域的软件的使用经历（编程，机械建模，艺术建模，多物理场仿真，电路仿真，光学仿真，...），还有解决问题的经验和耐性；梳理问题的逻辑性；等等。也许，这正是为什么我会如此惊讶于美赛的结果，因为明明没有去狠下功夫进行备赛，甚至五次比赛的备赛总时长不超过三个月。我惊讶于解决问题的能力在数学建模里如鱼得水，惊讶于数学建模比赛的本质并非数学建模，而是解决问题——拆解问题、约束并最终定义所有概念、解决一个个子问题。大道至简。

我不知道自己会在工科里呆多久，也并不确定自己要和技术打多久交道。

但，至少现在，这门学科看起来非常适合我。

Lucky
写于2024.5