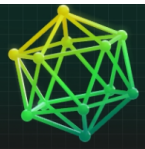
这里写一下我是如何准备复试的，申明一下，不涉及复试的细节，主要讲我自己是如何准备复试的。
专业面主要是 ai 方向
介绍一下复试，复旦的复试是英语面试+机试+专业面试或机试+英面+专业面，然后方向是专业面前或英面前晚上选。

首先是英语面试
基本内容：时长 10 分钟，自我介绍 1 分钟。
英语面试我是最后两周准备的，然后最后一周开始背。我准备了一个自我介绍，主要就是为了水时间，里面除了水字数的内容，提及了我是从自动化跨考的。然后我也准备了一些很简单的项目，bert 分类谣言，我也针对这个项目自己想了一些问题，可以利用准备的这些问题在考场上临时拼凑，就像背肖四一样，考场上不会出原题，但是有东西可以当做素材写。除此之外，我还准备了一些老生常谈的内容，为什么考复旦？你觉得复旦怎么样?你觉得上海怎么样？为什么选择考 cs？还有因为我是跨专业的，所以准备了相关的内容，为什么跨专业？原专业对 cs 有什么助？原专业学习了哪些内容？介绍原专业一门学的最好的课？除此之外还有其他经验贴提及的准备了数据结构的树的介绍，就准备了这一个。大家可以根据自己的特点然后结合自我介绍来准备。然后我背这些准备的内容，自我介绍和问题是死劲硬背大概背了 5 天，第一二天一直背，后面每天早晚背两遍巩固。然后我让我女朋友假装面试老师问我前面准备的问题。当然，英面不是最重要的，满分 10 分，大部分人都能拿 6 分以上。

接着是机试
机试的基本内容：可以选择 java c c++，时长好像是两个半小时，可以去官网看看。然后一共有 5 题。IDE 是 vscode dev vs，vscode 和 dev 可能不能用，有的机子可能没有 vs，看运气，所以最好三个 IDE 都熟悉，给分按测试通过的数据比例给分，样例只有几个，只会显示样例是否 accept。机试最好至少 a 两道以上，很多经验贴都写了机试还是挺重要的占 30 分，机试没做出来的题专业面也会问。
我的背景是：在考研前都没有接触过算法题，有编程基础，但是没有 leetcode 基础。
我是如何准备机试的：在考完研，对完答案心里大概有底了以后，就开始看 acwing 算法基础课了，当然刚开始也可以看代码随想录，看这个主要是为了自己准备代码模板以及了解一些基本题型。我自己准备了一些模板，https://github.com/BREKOJI/cs_test_template，开源给大家了，有需要自取。然后 acwing 的课不需要看数学那部分，其他都可以看，旦的机试爱考 dp，贪心，图论，然后每年都会有一题送分题（签到题）。我快速过完 acwing 后，马上开始做 leetcode，以及动态规划基础题，下面是我做了哪些题目，可以参考一下



更细的知识点拆分，让入门更简单 更多信息

动态规划（基础版）

38 / 50

显示标签

斐波那契类型

- 爬楼梯 简单
- 斐波那契数 简单
- 第 N 个泰波那契数 简单
- 使用最小花费爬楼梯 简单
- 打家劫舍 中等
- 删除并获得点数 中等

矩阵

- 不同路径 中等
- 最小路径和 中等
- 不同路径 II 中等
- 三角形最小路径和 中等
- 下降路径最小和 中等
- 最大正方形 中等

动态规划在字符串的应用

- 最长回文子串 中等
- 单词拆分 中等
- 最长回文子序列 中等
- 编辑距离 中等
- 两个字符串的最小ASCII删除和 中等
- 不同的子序列 困难

最长递增子序列

- 最长递增子序列 中等
- 最长递增子序列的个数 中等
- 最长数对链 中等
- 最长定差子序列 中等
- 最长等差数列 中等
- 俄罗斯套娃信封问题 困难
- 找出到每个位置为止最长的有效障碍赛跑路线 困难

最长公共子序列

- 最长公共子序列 中等
- 不相交的线 中等
- 让字符串成为回文串的最少插入次数 困难

买卖股票的最佳时间/状态机

- 买卖股票的最佳时机含冷冻期 中等
- 买卖股票的最佳时机含手续费 中等
- 买卖股票的最佳时机 III 困难
- 买卖股票的最佳时机 IV 困难

动态规划在树的应用

- 不同的二叉搜索树 中等
- 不同的二叉搜索树 II 中等
- 打家劫舍 III 中等
- 二叉树中的最大路径和 困难

背包问题

- 完全平方数 中等
- 零钱兑换 II 中等
- 组合总和 IV 中等
- 一和零 中等

动态规划 - 一维

- 解决智力问题 中等
- 零钱兑换 中等
- 统计构造好字符串的方案数 中等
- 解码方法 中等
- 最低票价 中等
- 多米诺和托米诺平铺 中等

动态规划 - 多维

升级 Plus 会员

力扣最受欢迎发烧友欢迎的 100 道题 更多信息

LeetCode 热题 100

68 / 100

显示标签

哈希

- 两数之和 简单
- 字母异位词分组 中等
- 最长连续序列 中等

双指针

- 移动零 简单
- 盛最多水的容器 中等
- 三数之和 中等
- 接雨水 困难

滑动窗口

- 无重复字符的最长子串 中等
- 找到字符串中所有字母异位词 中等

子串

- 和为 K 的子数组 中等
- 滑动窗口最大值 困难
- 最小覆盖子串 困难

普通数组

- 最大子数组和 中等
- 合并区间 中等
- 轮转数组 中等
- 除自身以外数组的乘积 中等
- 缺失的第一个正数 困难

矩阵

- 矩阵置零 中等
- 螺旋矩阵 中等
- 旋转图像 中等
- 搜索二维矩阵 II 中等

链表

- 相交链表 简单
- 反转链表 简单
- 回文链表 简单
- 环形链表 简单
- 环形链表 II 中等
- 合并两个有序链表 简单
- 两数相加 中等
- 删除链表的倒数第 N 个结点 中等
- 两两交换链表中的节点 中等
- K 个一组翻转链表 困难
- 随机链表的复制 中等
- 排序链表 中等
- 合并 K 个升序链表 困难
- LRU 缓存 中等

二叉树

- 二叉树的中序遍历 简单
- 二叉树的最大深度 简单
- 翻转二叉树 简单
- 对称二叉树 简单
- 二叉树的直径 简单
- 二叉树的层序遍历 中等
- 将有序数组转换为二叉搜索树 简单
- 验证二叉搜索树 中等
- 二叉搜索树中第 K 小的元素 中等
- 二叉树的右视图 中等
- 二叉树展开为链表 中等
- 从前序与中序遍历序列构造二叉树 中等
- 路径总和 III 中等
- 二叉树的最近公共祖先 中等
- 二叉树中的最大路径和 困难

图论

- 岛屿数量 中等
- 腐烂的橘子 中等
- 课程表 中等
- 实现 Trie (前缀树) 中等

回溯

- 全排列 中等
- 子集 中等
- 电话号码的字母组合 中等
- 组合总和 中等
- 括号生成 中等
- 单词搜索 中等
- 分割回文串 中等
- N 皇后 困难

二分查找

- 搜索插入位置 简单
- 搜索二维矩阵 中等
- 在排序数组中查找元素的第一个和最后一个位置 中等
- 搜索旋转排序数组 中等
- 寻找旋转排序数组中的最小值 中等
- 寻找两个正序数组的中位数 困难

栈

- 有效的括号 简单
- 最小栈 中等
- 字符串解码 中等
- 每日温度 中等
- 柱状图中最大的矩形 困难

堆

- 数组中的第K个最大元素 中等
- 前 K 个高频元素 中等
- 数据流的中位数 困难

贪心算法

- 买卖股票的最佳时机 简单
- 跳跃游戏 中等
- 跳跃游戏 II 中等
- 划分字母区间 中等

动态规划

- 爬楼梯 简单
- 杨辉三角 简单
- 打家劫舍 中等
- 完全平方数 中等
- 零钱兑换 中等
- 单词拆分 中等
- 最长递增子序列 中等
- 乘积最大子数组 中等
- 分割等和子集 中等
- 最长有效括号 困难

多维动态规划

- 不同路径 中等
- 最小路径和 中等
- 最长回文子串 中等
- 最长公共子序列 中等
- 编辑距离 中等

技巧

- 只出现一次的数字 简单
- 多数元素 简单
- 颜色分类 中等
- 下一个排列 中等
- 寻找重复数 中等

Hot100 和动态规划基础做的差不多了，就开始在动态规划题库里面找中等题做，其中穿插着之前做的 hot100 和动态规划基础题反复做。总共题数和进度如下：

做的差不多了以后，再整理一些输入输出的模板：比如如何根据数组建完全二叉树，构建哈夫曼树，dijstra 等等。可以参考 acwing 上面的往年题，来思考哪些类型的题的输入输出该准备。

一些前辈的模板可以参考：

https://www.acwing.com/blog/content/29816/

https://www.acwing.com/blog/content/35907/

https://www.acwing.com/user/myspace/record/264482/

最后是专业面

专业面是最重要的，占 60 分。接下来我将介绍我是如何准备 ai 的专业面的，也给出我的一些准备建议。

我的背景：我从大二下学期开始接触深度学习，之后参加了很多 kaggle、天池、讯飞等等的 ai 算法比赛，主要是 nlp 或多模态的。

我感觉我的经历可能对大多数同学没有参考价值，我专业面准备的比较少，主要是根据我自己的简历和自我介绍准备了可能问的问题。

不过下面我也给出我的一些建议（各个组的组面应该也可以这样子学，基本功扎实），

可以早点开始准备专业面，因为 ai 方向是不想要你就不及格，想要你就及格往上一点，所以主要还是看专业面发挥。可以早点准备，先看李宏毅的机器学习入门，做李宏毅的作业。然后差不多就入门了，接着可以做一些项目，这边推荐一个我做过的项目。

https://github.com/jingyaogong/minimind。也开源一下我写的注释

https://github.com/BREKOJI/minimind_note

Datawhale 的复现 qwen 的项目也可以，

https://github.com/datawhalechina/tiny-universe

选择一个项目，把代码全部看懂，把理论熟悉一下，流程跑一遍，有时间自己手搓一遍，就可以了。

可以通过下面这个考察一下自己知识的深度，提高自己对理论的理解：

对于理论方面，我准备了与我自己简历相关的一些内容:

Xgb lgb cat，这三个是什么，改进和优点？

GBT（gradient boosting tree）是基于决策树的算法，boosting 的思想，逐步训练多个弱决策树模拟残差。

这三个都是对 GBT 的改进

Xgb 最主要的是加入正则化和并行计算

Lgb 最主要的是使用直方图计算和梯度采样以及 leaf-wise 深度限制

Cat 是尽量构建对称决策树并且有序（数据集有序）训练减少过拟合

决策树是什么？

1. 决策树的核心思想

决策树的核心思想是：

- **递归划分**：通过选择最优特征和划分点，将数据集逐步划分为更小的子集，直到满足停止条件。
- **树形结构**：最终生成一棵树，其中每个内部节点表示一个特征划分，每个叶节点表示一个预测结果。
- **规则导向**：决策树的预测过程可以看作是一系列规则的组合，易于理解和解释。

SVM 是什么？

支持向量机（Support Vector Machine, SVM） 是一种经典的**监督学习算法**，主要用于分类和回归任务。SVM 的核心思想是找到一个**最优超平面**，将不同类别的数据分开，并最大化类别之间的**间隔（Margin）**。由于其强大的分类能力和理论基础，SVM 在机器学习领域得到了广泛应用。

以下是对 SVM 的详细介绍：

1. SVM 的核心思想

SVM 的核心思想是：

- **最优超平面**：在特征空间中找到一个超平面，将不同类别的数据分开。
- **最大化间隔**：选择能够最大化两类数据之间间隔的超平面，从而提高模型的泛化能力。
- **支持向量**：距离超平面最近的样本点称为支持向量，它们决定了超平面的位置和方向。

机器学习模型（xgb、lgb、cat）与大模型结果集成，为什么这么做？

对结构化数据或规则性模式的建模能力有限，而 xgb、lgb、cat 等机器学习模型在这方面有优势。

Roberta 和 bert 有什么区别？

预训练使用了更大的数据集和 batch_size，预训练使用动态掩码（bert 是静态掩码），bert 是 next-token prediction 和 mask lm 都是用，roberta 是只用 mask lm

OCR 用的什么

Paddlepaddle 的 ocr 库

GPTQ 是什么？

将浮点数 scale 到整数级别，然后再舍入小数，来减小显存占用，提高运算速度，丢一点精度。

Multi-head-attention 和 self-attention

MHA 分成多个头，然后将特征维度均分给每个头处理，每个头都是 Q、K、V 矩阵计算注意力得分，最后再拼接结果。

为什么文档检索需要分两个不同的块？

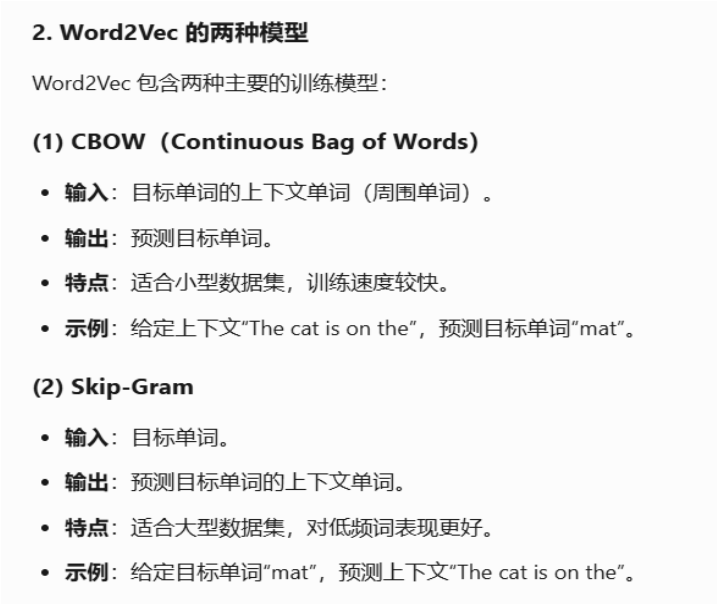
Pdf 分为长的和短的分块，长的保留语义信息，短的可以捕捉细节。

bge、gte、bm25
多路召回，bge 和 gte 擅长理解上下文，然后 bm25 是基于词频逆文档的，擅长捕捉关键词，粗排之后再精排一次。

逻辑回归和分类：
逻辑回归一般用于二分类问题，最后是 sigmoid，输出概率是连续的，但是一般会取 0.5 为阈值分类，分类是回归得到概率对概率进行采样选择。

逻辑回归和 softmax:
逻辑回归一般用于二分类问题，sigmoid 之后为 0 到 1 区间。softmax 是处理多分类问题，有点像归一化，让向量的概率指数总和为 1，然后采样进行分类。

Word2vec 是什么



2. Word2Vec 的两种模型

Word2Vec 包含两种主要的训练模型：

(1) CBOW（Continuous Bag of Words）

- **输入**：目标单词的上下文单词（周围单词）。
- **输出**：预测目标单词。
- **特点**：适合小型数据集，训练速度较快。
- **示例**：给定上下文“The cat is on the”，预测目标单词“mat”。

(2) Skip-Gram

- **输入**：目标单词。
- **输出**：预测目标单词的上下文单词。
- **特点**：适合大型数据集，对低频词表现更好。
- **示例**：给定目标单词“mat”，预测上下文“The cat is on the”。

Word2vec 是 embedding 的一种
上面的几种方法，可以分为两大类，一类是 count-based，另一类是 prediction-based
Skip-gram 和 cbow 是 prediction-based，glove 是 count-based 和 prediction-based 结合
Multi head latent attention
是把输入 X 先经过线性层映射为 Z，再把 Z 经过 qkv 矩阵得到 QKV
其他的 attention 是直接把 X 经过 qkv 矩阵得到 qkv

Vllm
并发之类的来提升速度

TF-IDF 是什么？

2. TF-IDF 的计算公式

TF-IDF 的计算公式如下：

(1) 词频（TF）

词频表示一个词在文档中出现的频率。常见的计算方法有两种：

- **原始词频**：

$$\mathrm{TF}(t,d) = \frac{\text{词 } t \text{ 在文档 } d \text{ 中出现的次数}}{\text{文档 } d \text{ 中的总词数}}$$

- **对数词频**：

$$\mathrm{TF}(t,d) = \log(1 + \text{词 } t \text{ 在文档 } d \text{ 中出现的次数})$$

(2) 逆文档频率（IDF）

逆文档频率表示一个词在文档集合中的稀有程度。计算公式为：

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{文档集合 } D \text{ 中的总文档数}}{\text{包含词 } t \text{ 的文档数}}\right)$$

- 如果一个词在所有文档中都出现，其 IDF 值为 0。
- 如果一个词在很少的文档中出现，其 IDF 值较高。

(3) TF-IDF

TF-IDF 是 TF 和 IDF 的乘积：

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

计算词频和词稀有度的乘积

Tokenizer

bpe 用合并的方法构建词表

多模态任务中，文本和图像特征对齐是个难题，你有什么想法解决?

可以用对比学习构建正负样本对或者跨模态注意力机制

跨模态注意力机制，就是把图片分割成多个向量当做 token 和文本一起输入

Qwen2.5-instruct-GPTQ-Int8：

GQA

GPTQ 量化压缩精度

DPO

更大的训练数据集

GRPO：

用对比数据集训练奖励模型来预测需要对齐的模型的好坏，一般这种奖励模型可以选择同样的大语言模型，然后再模型生成过程中，给予奖励和惩罚，最大化好输出和坏输出奖励差异，对坏输出施加惩罚，抑制坏输出。

1. 目标函数的组成

目标函数由两部分组成：

1. **奖励部分**：$\log \sigma(r(y_w) - r(y_l))$
2. **惩罚部分**：$\lambda \cdot \text{Penalty}(y_l)$

整体目标函数为：

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(r(y_w) - r(y_l)) - \lambda \cdot \text{Penalty}(y_l)\right]$$

2. 符号说明

- x：输入提示（Prompt）。
- y_w：高质量输出（Preferred Output）。
- y_l：低质量输出（Dispreferred Output）。
- $\mathcal{D}$：数据集，包含输入 x 及其对应的高质量输出 y_w 和低质量输出 y_l。
- $r(y_w)$ 和 $r(y_l)$：奖励模型对高质量输出和低质量输出的评分。
- σ：Sigmoid 函数，$\sigma(z) = \frac{1}{1+e^{-z}}$。
- $\text{Penalty}(y_l)$：对低质量输出的惩罚函数。
- λ：惩罚权重，用于平衡奖励和惩罚。

3. 奖励部分：$\log \sigma(r(y_w) - r(y_l))$

- **目标**：最大化高质量输出 y_w 和低质量输出 y_l 之间的奖励差异。
- **解释**：
  - $r(y_w) - r(y_l)$：奖励模型对高质量输出和低质量输出的评分差异。
  - $\sigma(r(y_w) - r(y_l))$：通过 Sigmoid 函数将评分差异映射到 (0, 1) 区间。
  - $\log \sigma(r(y_w) - r(y_l))$：对 Sigmoid 值取对数，最大化该值意味着最大化评分差异。
- **作用**：鼓励模型生成更多高质量输出，减少低质量输出。

4. 惩罚部分：$\lambda \cdot \text{Penalty}(y_l)$

- **目标**：对低质量输出 y_l 施加惩罚，抑制模型生成类似的输出。

DPO、GQA、MHLA、RoPE、BPE、moe、白盒蒸馏、黑盒蒸馏

我准备的内容，可以作为一些参考，不过最后还是根据你自己的简历来准备，我这些准备的是和我的简历相关的。另外老师问问题考察有一定深度（打破砂锅问到底）压力面，需要你对理论有自己的思考。
时间安排：我是在考完研之后一个月准备了最后一个项目就是前面的 minimind，然后复试前一周准备了专业面简历和一些问题。专业面前一天晚上回顾理论和项目。
专业面总时长 15 分钟，自我介绍大概 1 到 2 分钟。

差不多就这些了。