



李谕杰

Java后端开发, 测试开发
北京
jacksongrayjackson24@gmail.com
18813158769

<https://github.com/CroCooCn>
OI算法竞赛blogs
<https://juejin.cn/user/2235744847674059>

教育背景

北京交通大学211 - 信息安全 - 本科

2024.09 - 至今

实习经历

北京京天威科技发展有限公司 - Java后端开发实习生

2025.06 - 2025.09

- 落地数据接入平台 (**Spring Boot + Kafka**)，统一接入5类车统数据与2类设备报文，标准化REST接口输出；支持日均**50万+**、峰值稳定**300条/s**，单实例吞吐**180→260条/s (+44%)**，连续稳定运行**90天**0起P0/P1。
- 实现幂等入库链路 (**FastJSON2/Jackson解析 + MyBatis-Plus基于业务键去重**)，将重复入库率从**2.8%→0.3% (-89%)**，错误重试/回滚减少约**40%**，显著提升一致性与可追溯性。
- 设计持久层与批量入库，结合**动态数据源 + Druid连接池**支撑高并发稳定写入；并发场景下P95接口时延**360→235ms (-35%)**，P99 SQL**1.6→1.1s (-31%)**，数据库超时/阻塞告警下降约**60%**。

项目经历

AI朋友圈

2025.09 - 2025.10

SpringBoot WebFlux MybatisPlus MySQL Maven

- 项目简介**：朋友圈图文内容AI生成与管理系统，提供手机号验证码登录、点券结算、AI 图生文案、图片优化渲染、素材上传与生成历史管理等能力。
- 项目内容**：
 - 提示词模板化+可插拔大模型：外部文件储存带变量的提示词模板便于修改，根据用户上传的素材类型，自动切换text/vision/thinking模型。
 - 图文闭环：可根据用户上传的图片生成结构化朋友圈文案，也可根据文案优化图片，支持用户多轮对话文案修改建议，为每轮对话保留上下文，根据使用场景分类保存AI交互历史。
 - 强鲁棒性：用正则表达式解析模型输出，若解析失败，自动切换其他模型重新尝试，图片渲染功能配有本地图片渲染工具作为兜底方案。

仿大众点评

2025.03 - 2025.04

SpringBoot Redis MybatisPlus Hutool MySQL Maven

- 项目简介**：实现店铺与优惠券、探店笔记、用户关注/点赞/推送、抢券秒杀等核心业务，配套多种登录鉴权与多种并发场景下的应对策略，通过配置轻松切换。覆盖高并发、热点数据、地理位置服务等典型场景。
- 项目内容**：
 - 秒杀券抢购与并发控制：使用 Lua 脚本原子化校验“库存充足+一人一单”，支持四种秒杀下单策略切换：悲观锁、乐观锁、内存阻塞队列异步落库、Redis Stream 异步落库。
 - 缓存体系与热点治理：使用存储空值防止缓存穿透，使用互斥锁或逻辑过期防止缓存击穿，更新采用“先改库后删缓存”的 Cache-Aside 模式。
 - 社交与内容分发：探店笔记点赞用 ZSet 存 TopN 点赞用户；采取推模式，发布笔记时将 BlogID 推送到粉丝的 feed ZSet。

专业技能

- 熟练掌握 **Java 核心知识体系**，理解面向对象、集合与常用数据结构、反射机制、类加载过程及异常体系等。
- 熟练掌握 **MySQL**，深入理解事务原理、索引与锁机制、MVCC、存储引擎及多种日志体系，具备数据库性能优化思维。
- 熟练掌握 **Redis 缓存体系**，重点解决缓存穿透、缓存击穿、缓存雪崩等问题，理解常用数据结构、持久化策略及高可用方案。
- 熟悉 **JUC 并发编程体系**，掌握线程池、AQS、CAS、锁机制、ThreadLocal 等核心组件及底层实现原理，有一定并发性能优化经验。
- 熟练使用 **SSM、Spring Boot、Spring Cloud** 等主流框架，深入理解 IOC、AOP、Bean 生命周期与循环依赖处理机制。
- 熟练使用 **Kafka**，掌握分区与副本机制、消费组负载均衡、消息堆积处理策略及高吞吐实践经验。

个人荣誉

蓝桥杯程序设计竞赛大学生A组省二等奖

2025

CACC第一届中国计算机学会算法能力大赛省三等奖

2024

Wait——Really Boring Like THAT!?

我选择马上动手，不是因为它简单，正是因为它有挑战！

热爱算法 (NOT ai Algorithm BUT OI Algorithm)

- 暑假中，曾用两周时间，一心探究广义下取整和算法（一种类欧几里得算法），最终发现，递归层数相同，一些参数会相同，因此使用记忆化的方式优化程序性能！

文章 Link:

<https://juejin.cn/post/7520687447236804671?searchId=2025112020083532635DA60F713FFE0DBA>

广义下取整和: $\sum_{i=0}^{N-1} i^p \lfloor \frac{Ai+B}{M} \rfloor^q = f(N, M, A, B)^{p, q}$.

其中 A, B, M, N 是整数, p, q 是大于等于 0 的整数.

在进行推导前, 先引入: $S(p, N) = \sum_{k=0}^{N-1} k^p = \sum_{i=0}^{p+1} S(p, i) N^i$.

给定 p, i , 就确定了 $S(p, i)$.

$S(p, N)$ 可用 N 时间用第一个式子算出, 或在已知 $S(p, i)$ 的情况下, 用 p 时间用第二个式子算出.

推导: ① 先归纳: $A = A_1 M + A_2, B = B_1 M + B_2$.

$$\begin{aligned} i^p \lfloor \frac{Ai+B}{M} \rfloor^q &= i^p \lfloor \frac{(A_1 M + A_2)i + B_1 M + B_2}{M} \rfloor^q \\ &= i^p \lfloor \frac{A_1 i + B_1 + \lfloor \frac{A_2 i + B_2}{M} \rfloor}{1} \rfloor^q \\ &= i^p \sum_{j=0}^{\lfloor \frac{A_2 i + B_2}{M} \rfloor} (A_1 i + B_1 + j)^q \cdot \lfloor \frac{A_2 i + B_2}{M} \rfloor^{q-j} \cdot \binom{q}{j} \\ &= i^p \sum_{j=0}^q \sum_{t=0}^j (A_{11})^t \cdot B_1^{j-t} \cdot \lfloor \frac{A_2 i + B_2}{M} \rfloor^{q-j} \cdot \binom{q}{j} \binom{j}{t} \\ &= \sum_{j=0}^q \sum_{t=0}^j \binom{q}{j} \binom{j}{t} A_1^t B_1^{j-t} \cdot i^{p+t} \lfloor \frac{A_2 i + B_2}{M} \rfloor^{q-j} \end{aligned}$$

再加上 $\sum_{i=0}^{N-1}$, 则 $f(N, M, A, B)^{p, q}$

$$= \sum_{j=0}^q \sum_{t=0}^j \binom{q}{j} \binom{j}{t} A_1^t B_1^{j-t} \cdot f(N, M, A_2, B_2)^{p+t, q-j}$$

② 计算 $f(N, M, A, B)^{p, q}$. 其中 $A, B \in [0, M)$.

同样地, 让 $\lfloor \frac{Ai+B}{M} \rfloor$ 可以取 $0, 1, 2, \dots, \lfloor \frac{A(N-1)+B}{M} \rfloor = k$.

$$k \leq \frac{A(N-1)+B}{M} < k+1$$
$$\frac{A}{M} d_k = \lfloor \frac{kM+B}{A} \rfloor, \forall i \in [d_k, d_{k+1})$$

且 $d_{k+1} = N$.

$$\sum_{k=0}^K \left((0^p + 1^p + 2^p + \dots + (d_{k+1}-1)^p) - (0^p + 1^p + \dots + (d_k-1)^p) \right) \cdot k^q$$

i^p (当 k 所取的 i).

$$= \sum_{k=0}^K (S(p, d_{k+1}) - S(p, d_k)) \cdot k^q$$

FF4
掘金技术社区 @ LeoBlues

```
C++
inline ll gfs(ll n, ll m, ll a, ll b, ll p, ll q, ll dep){
    if(memo[dep][p][q] != -1) {
        return memo[dep][p][q];
    }
```

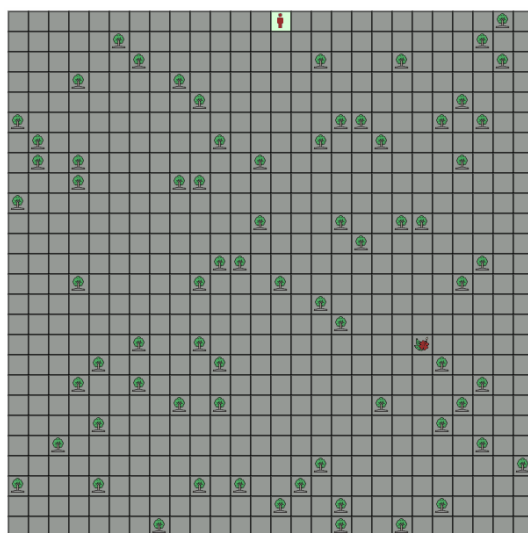
```

    }
    if(n<=0) return memo[dep][p][q]=0;
    if(q==0) return memo[dep][p][q]=get_S(p,n);
    if(a>=m || b>=m || b<0) { //归约步骤
        ll a1=a/m,b1=b/m;
        ll a2=a%m;
        ll b2=b%m;
        ll ans=0;
        for(ll j=0;j<=q;j++){
            for(ll t=0;t<=j;t++){
                ans=(ans+com(q,j)*com(j,t)%mod*ipow(a1,t)%mod*ipow(b1,j-t)%mod*gfs(n,m,a2,b2,p+t,q-j,dep+1)%mod)%mod;
            }
        }
        memo[dep][p][q]=ans;
        return ans;
    }else { //交换步骤
        ll K=(a*(n-1)+b)/m;
        ll ans=ipow(K,q)*get_S(p,n)%mod;
        if(a<=0) return memo[dep][p][q]=ans;
        ll delta=0;
        for(ll i=0;i<=q-1;i++){
            for(ll j=0;j<=p+1;j++){
                if(s[p][j]!=0){ //因很多 s[x][y]=0, 故进行此优化终止递归
                    ll t=gfs(K,a,m,m-b+a-1,i,j,dep+1);
                    delta=(delta+com(q,i)*t%mod*s[p][j]%mod)%mod;
                }
            }
        }
        delta=delta*invden[p]%mod;
        ans=(ans-delta)%mod;
        memo[dep][p][q]=ans;
        return ans;
    }
    return 0; //guard
}

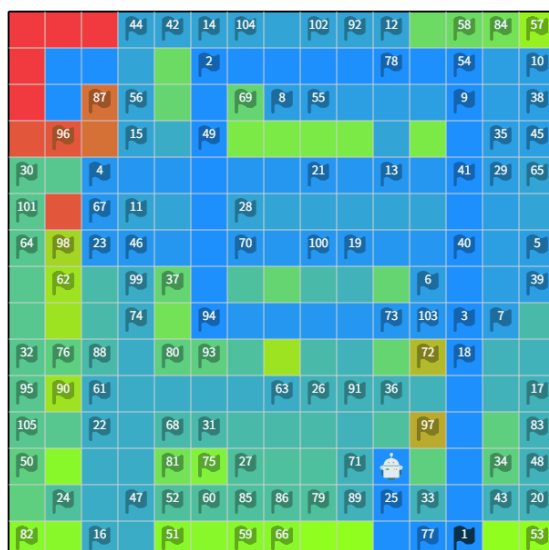
```

- 在不会任何启发式算法的情况下，报名参加 Atcoder 的启发式算法马拉松，利用 **10** 天时间，迭代近 **10** 个版本，取得 **390/1100** 的名词！

在 ChatGPT 的辅助下，利用**集束搜索**等启发式方法，实时布置动态迷宫：



- 机器人路径规划！研究了数天分组优化的方法，但发现行不通，虽然最终只提交了第一版本的程序，但结果依然不错！



开发——AIGC, Dev to the FUTURE！

- AI 朋友圈（独立开发） **Java+React+SQL**



- AI 命理预测（合作开发）Java+Js+MongoDB



- 参加 Google DevFest 2025

Vibe Coding 制作了一个关于 MCP 使用心得的网站：<https://crocoocn.github.io/DevFest2025Beijing-Web/index.html>



未来——珍惜机会，分享点子，迎接挑战！