定时刷新

沿用诛仙逻辑

top排行



* 数量不是很大，每次请求把名单全部发给客户端，如果需要对该榜进行筛选，客户端去筛选（筛选是在已有名单里筛选）
* 同时使用两个时间戳，客户端记获取榜单的时间戳，服务器记更新榜单的时间戳，T间隔内最多发送一次榜单
* 实现

struct RankInfo

{

int key;

int score;

int other\_info;

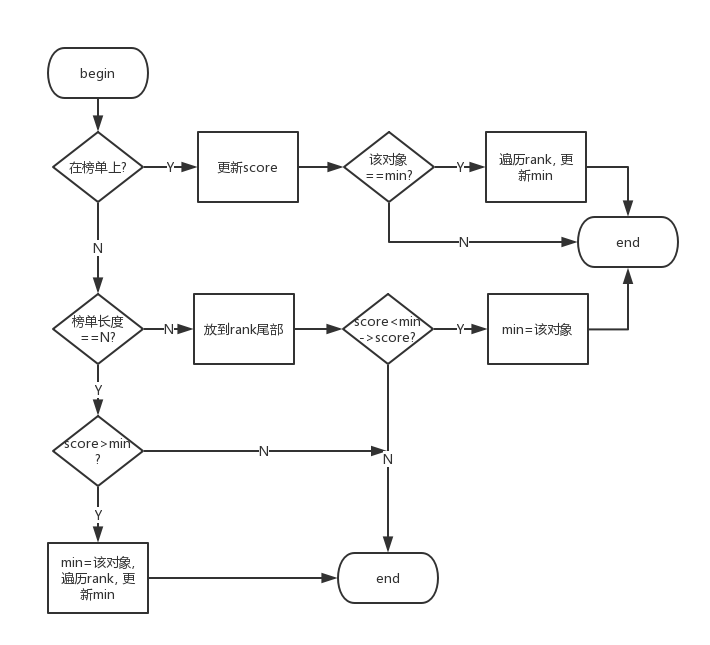
int last\_ranking;

};

RankInfo rank[N];

RankInfo \*min;

hashmap<int, RankInfo\*> rankMap;



为了获得排名变化，服务器定时进行排序（比如10分钟一次），变化的少的话使用插入排序，变化的多用快排

* 其他问题

避免频繁的榜单变化（如两人经验相近情况下挂机导致的等级排行榜频繁变化），为这种可能会频繁变化的榜单设置一个计时或计数，或者弄一个缓存，同一玩家的数据累计到特定次数或特定时间才更新

对于不需要筛选功能的榜单，因为排好序的榜单是存在数组里的，如果榜单比较大，可以很容易实现分页传输

存储：定时存储，停服重刷

全排

比如现在的鲜花榜

与上面相比，没有了数量的限制，实际上相当于一个很大的N

将score分割成多个桶，每个桶内部使用类似上面的方法，同时另外维护一个<key,score>的map，来找出对象属于哪个桶

或者用当前莲台排行榜/鲜花榜的方法，使用以score为key的map，每次发生变化删除旧分数，添加新分数

获取个人排名

榜单之上的准确排名，榜单之外的近似排名

离线获取所有成员的排名分布，比如等级排行榜，分别统计每个等级有多少人，给出一个级别，就能获得大概有多少人等级更高。

问题：排名分布不能实时更新

搜索功能

模糊搜索

如果能在内存中放下所有数据，有可能在内存中通过子串搜索进行搜索，否则最好是借助其他工具，比如腾讯云搜，百度对话式搜索BCS