

牛人计划-高级项目课（9）



牛客网
NOWCODER



第九课

课程目录

CONTENTS

- 关注/被关注服务
- 关注问题
- 关注人
- 排序算法介绍



关注服务（代码实现）

概念：

1. A关注B
2. A是B的粉丝（follower）
3. B是A的关注对象（followee）

特点：

1. 多对多服务
2. ID和ID的关联，有序

存储结构：

redis：zset/list

Service：

1. 通用关注接口
2. 粉丝列表分页
3. 关注对象列表分页

Controller：

1. 首页问题关注数
2. 详情页问题关注列表
3. 粉丝/关注人列表
4. 关注异步事件



redis事务

```
try {  
    Transaction tx = jedis.multi();  
    tx.zadd("qq", 2, "1");  
    tx.zadd("xx", 3, "2");  
    List<Object> objs = tx.exec();  
    tx.close();  
} catch (Exception e) {  
    e.printStackTrace();  
} finally {  
}
```



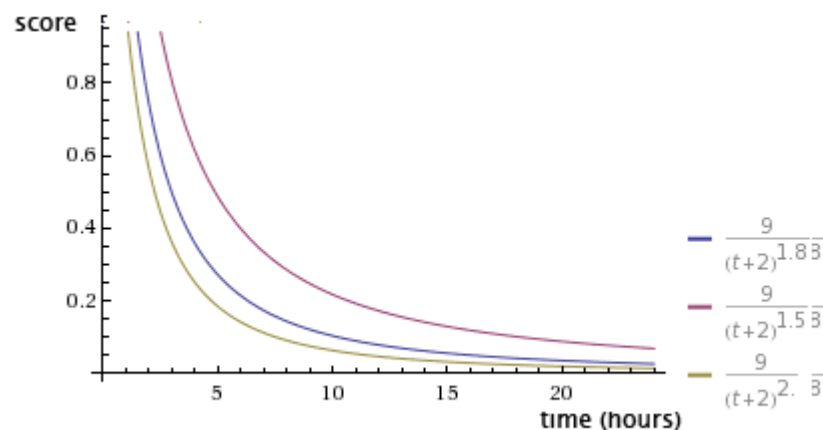
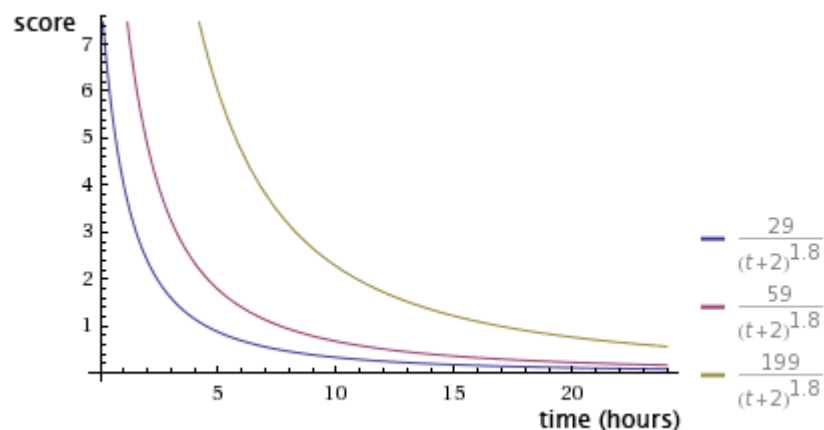
Hacker News <https://news.ycombinator.com/>

$$\text{Score} = (P - 1) / (T + 2)^G$$

P：投票数，-1是把自己投的过滤掉

T：发布到现在的时间间隔，单位小时，+2防止除数太小

G：重力加速度，分值根据时间降低速率



<https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d>



Reddit <https://www.reddit.com/>

Given the time the entry was posted A and the time of 7:46:43 a.m. December 8, 2005 B , we have t_s as their difference in seconds

$$t_s = A - B$$

and x as the difference between the number of up votes U and the number of down votes D

$$x = U - D$$

where $y \in \{-1, 0, 1\}$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and z as the maximal value, of the absolute value of x and 1

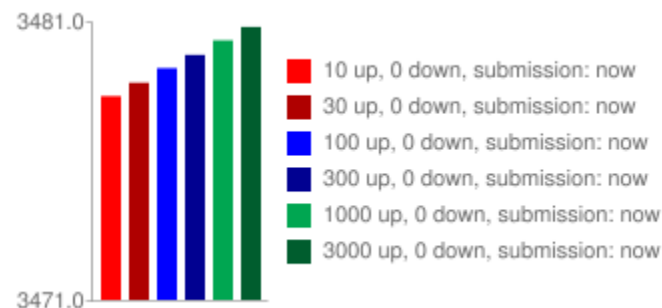
$$z = \begin{cases} |x| & \text{if } |x| \geq 1 \\ 1 & \text{if } |x| < 1 \end{cases}$$

we have the rating as a function $f(t_s, y, z)$

$$f(t_s, y, z) = \log_{10} z + \frac{y t_s}{45000}$$

$86400/45000=1.92$ 一天权重调整
 $10^{1.92}=83$ 投票差要涨83倍

赞成票加分，投票差前10和接下来的100等权



时间是最重要的权重，由于流量比较大，所以对于高赞文章有所优势，适合新闻类排序

<https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9>



NOWCODER.COM
牛客网-中国最大IT笔试/面试题库

StackOverflow <http://stackoverflow.com>

$$\frac{(\log(Qviews)*4) + ((Qanswers * Qscore)/5) + \text{sum}(Ascores)}{((QageInHours+1) - ((QageInHours - Qupdated)/2)) ^ 1.5}$$

Qviews : 问题浏览数，通过log来平滑

Qanswer : 问题回答数，有回答的题目才是好问题

Qscore : 问题赞踩差，赞的越多，问题越好

sum (Ascores) : 回答赞踩差，回答的越多问题越好

QageInHours : 题目发布时间差，时间越久排名越后

Qupdated : 最新的回答时间，越新关注度越高

<http://meta.stackexchange.com/questions/11602/what-formula-should-be-used-to-determine-hot-questions>



IMDB <http://www.imdb.com/chart/top>

加权排名 (WR) = $(v \div (v+m)) \times R + (m \div (v+m)) \times C$

R = 某电影投票平均分

v = 有效投票人数

m = 最低投票人数, 1250

C = 所有电影平均值

投票人数越多, 越偏向于用户打分值, 防止冷门电影小数人高分导致的高分

http://www.imdb.com/help/show_leaf?votestopfaq

<https://community.topcoder.com/longcontest/?module=Static&d1=support&d2=ratings>



课后练习

1. Profile页增加显示共同关注
2. 自定义排序算法实现



Thanks