



各种字符串Hash函数比较

常用的字符串Hash函数还有ELFHash，APHash等等，都是十分简单有效的方法。这些函数使用位运算使得每一个字符都对最后的函数值产生 影响。另外还有以MD5和SHA1为代表的杂凑函数，这些函数几乎不可能找到碰撞。

常用字符串哈希函数有 BKDRHash，APHash，DJBHash，JSHash，RSHash，SDBMHash，PJWHash，ELFHash等等。对于以上几种哈 希函数，我对其进行了一个小小的评测。

Hash函数	数据1	数据2	数据3	数据4	数据1得分	数据2得分	数据3得分	数据4得分	平均分
BKDRHash	2	0	4774	481	96.55	100	90.95	82.05	92.64
APHash	2	3	4754	493	96.55	88.46	100	51.28	86.28
DJBHash	2	2	4975	474	96.55	92.31	0	100	83.43
JSHash	1	4	4761	506	100	84.62	96.83	17.95	81.94
RSHash	1	0	4861	505	100	100	51.58	20.51	75.96
SDBMHash	3	2	4849	504	93.1	92.31	57.01	23.08	72.41
PJWHash	30	26	4878	513	0	0	43.89	0	21.95
ELFHash	30	26	4878	513	0	0	43.89	0	21.95

其中数据1为100000个字母和数字组成的随机串哈希冲突个数。数据2为100000个有意义的英文句子哈希冲突个数。数据3为数据1的哈希值与 1000003(大素数)求模后存储到线性表中冲突的个数。数据4为数据1的哈希值与10000019(更大素数)求模后存储到线性表中冲突的个数。

经过比较，得出以上平均得分。平均数为平方平均数。可以发现，BKDRHash无论是在实际效果还是编码实现中，效果都是最突出的。APHash也 是较为优秀的算法。DJBHash,JSHash,RSHash与SDBMHash各有千秋。PJWHash与ELFHash效果最差，但得分相似，其算 法本质是相似的。

```

unsigned int SDBMHash(char *str)
{
    unsigned int hash = 0;

    while (*str)
    {
        // equivalent to: hash = 65599*hash + (*str++);
        hash = (*str++) + (hash << 6) + (hash << 16) - hash;
    }

    return (hash & 0x7FFFFFFF);
}

// RS Hash Function
```

搜索

我的标签

随笔分类

- Algorithm(10)
- Chess
- Compiler(1)
- Data structure(5)
- DirectX(2)
- Fractal(2)
- Graphics(13)
- Mathematica
- Mathematics(15)
- OpenCL
- OpenGL(7)
- Optimization(3)
- Puzzle(10)
- SSE(3)

随笔档案

- 2011年1月 (1)
- 2010年12月 (2)
- 2010年8月 (2)
- 2010年5月 (1)
- 2010年4月 (2)
- 2010年2月 (2)
- 2010年1月 (6)
- 2009年12月 (6)
- 2009年11月 (12)

最新评论XML

- 1. Re: 3D图形学(4) - 视空间变换
- 3. 我们在2得到绕任意轴的旋转矩阵，

```
unsigned int RSHash(char *str)
{
    unsigned int b = 378551;
    unsigned int a = 63689;
    unsigned int hash = 0;

    while (*str)
    {
        hash = hash * a + (*str++);
        a *= b;
    }

    return (hash & 0x7FFFFFFF);
}

// JS Hash Function
unsigned int JSHash(char *str)
{
    unsigned int hash = 1315423911;

    while (*str)
    {
        hash ^= ((hash << 5) + (*str++) + (hash >> 2));
    }

    return (hash & 0x7FFFFFFF);
}

// P. J. Weinberger Hash Function
unsigned int PJWHash(char *str)
{
    unsigned int BitsInUnsignedInt = (unsigned int)(sizeof(unsigned int) * 8);
    unsigned int ThreeQuarters = (unsigned int)((BitsInUnsignedInt * 3) / 4);
    unsigned int OneEighth = (unsigned int)(BitsInUnsignedInt / 8);
    unsigned int HighBits = (unsigned int)(0xFFFFFFFF) << (BitsInUnsignedInt - OneEighth);
    unsigned int hash = 0;
    unsigned int test = 0;

    while (*str)
    {
        hash = (hash << OneEighth) + (*str++);
        if ((test = hash & HighBits) != 0)
        {
            hash = ((hash ^ (test >> ThreeQuarters)) & (~HighBits));
        }
    }

    return (hash & 0x7FFFFFFF);
}

// ELF Hash Function
unsigned int ELFHash(char *str)
{
    unsigned int hash = 0;
    unsigned int x = 0;

    while (*str)
    {
        hash = (hash << 4) + (*str++);
        if ((x = hash & 0xF000000L) != 0)
        {
            hash ^= (x >> 24);
            hash &= ~x;
        }
    }
}
```

问题: 是否任意旋转矩阵M都围绕某一轴? 答案是否定的, 设e1, e2, e3是单位向量, s1 = M*e1, s2 = M*e2, s3 = M*e3 如果存在这样的轴l, 则 l = e1*s1 = e2*s2 = e3*s3, 只需要在这式子找一个反例即可 这个地方不太清楚, lz能给解释下不?

--ITvsET

* 阅读排行榜

- 1. 各种字符串Hash函数比较(2601)
- 2. 供智商超过180的人使用的测试卷: Haselbauer-Dickheiser Test(1696)
- 3. 3D 图形学(6) - 欧拉旋转之万向锁(Gimbal Lock)问题解释(1148)
- 4. 3D图形学(4) - 视空间变换(756)
- 5. 3D图形学(5) - 四元数(564)

* 评论排行榜

- 1. 3D图形学(4) - 视空间变换(2)
- 2. 64K 动画 技术剖析之: Metaball (1)
- 3. 3D图形学(7) - 光线跟踪-反射(1)
- 4. Which Way Did the Bicycle Go 趣题选 (上) (0)
- 5. 用选择公理来预测未来(0)

```
        return (hash & 0x7FFFFFFF);
    }

// BKDR Hash Function
unsigned int BKDRHash(char *str)
{
    unsigned int seed = 131; // 31 131 1313 13131 131313 etc..
    unsigned int hash = 0;

    while (*str)
    {
        hash = hash * seed + (*str++);
    }

    return (hash & 0x7FFFFFFF);
}

// DJB Hash Function
unsigned int DJBHash(char *str)
{
    unsigned int hash = 5381;

    while (*str)
    {
        hash += (hash << 5) + (*str++);
    }

    return (hash & 0x7FFFFFFF);
}

// AP Hash Function
unsigned int APHash(char *str)
{
    unsigned int hash = 0;
    int i;

    for (i=0; *str; i++)
    {
        if ((i & 1) == 0)
        {
            hash ^= ((hash << 7) ^ (*str++) ^ (hash >> 3));
        }
        else
        {
            hash ^= (~((hash << 11) ^ (*str++) ^ (hash >> 5)));
        }
    }

    return (hash & 0x7FFFFFFF);
}
```



<http://www.byvoid.com/blog/string-hash-compare/>

分类: [Algorithm](#)

绿色通道:

好文要顶

关注我

收藏该文

与我联系



[fairstar](#)

关注 - 0

2

0



粉丝 - 2

+加关注



(请您对文章做出评价)

- « 博主前一篇: [智力题: 数轴上的潜水艇](#)
- » 博主后一篇: [3D 图形学\(6\) - 欧拉旋转之万向锁\(Gimbal Lock\)问题解释](#)

posted @ 2010-02-06 06:56 fairstar 阅读(2601) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

程序员问答社区，解决您的IT难题
[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)

[阅读完整内容 >](#)

[cn.iforex.com](#) [Ads by Google](#)

- 最新IT新闻:
- [种族主义根源](#)
 - [苹果为数据中心提供清洁能源 建太阳能矩阵](#)
 - [亚马逊拟销售Kindle Fire 欢迎屏广告位](#)
 - [谷歌承诺5年内Android系统免费](#)
 - [Google 智能眼镜的输入方式](#)
- » [更多新闻...](#)

- 最新知识库文章:
- [12 个有效的提高编程技能的方法](#)
 - [用情感化交互为你的产品注入灵魂](#)
 - [简单之美——系统设计黄金法则](#)
 - [不懂技术的人不要对懂技术的人说这很容易实现](#)
 - [QQ空间技术架构之深刻揭秘](#)
- » [更多知识库文章...](#)

寻人启事

Android应用开发!
你是自由职业者?

china-pub.com

买一赠一
全场满48元包邮

China-Pub 低价书精选
China-Pub 计算机绝版图书按需印刷服务