



算法杂记 | 哈希算法



江南西路转运使
高二, Oler, 南宋史小白, 『ユイカ』粉丝

关注他

1人赞同了该文章

概念

哈希算法(Hash), 又称散列、映射, 是一种十分常用的算法, 广泛应用于字符串、加密等领域。本篇文章将着重介绍字符串哈希, 包括求哈希值、判断两字符串是否相等、以及二维哈希等等。

原理

一维哈希

每一个字符都有其ASCII码。我们把这些字符串当作一个个 *base* 进制数, 数值就是哈希值。例如对于字符串 *abcd* 的哈希值就是 $a * base^3 + b * base^2 + c * base^1 + d * base^0$ 。一般 *base* 值可取 131 或 13331 等质数。但是长字符串的哈希值显然会超过 *long long* 类型的数据范围, 因此我们还需要对哈希值取模, 模数一般采用 $10^9 + 7$ 或 $10^9 + 9$ 等大质数。之所以都取质数, 是因为可以尽量避免“哈希冲突”的发生, 即两个不相同的字符串的哈希值相等。

一般求字符串的哈希值都会求它的前缀哈希值, 并用 $hs[i]$ 存储 $[1, i]$ 的前缀的哈希值。代码如下

```
ll mo(ll u) {return ((u%mod+mod)%mod);}
...
for (int i=1; i<=n; i++) hs[i]=mo(hs[i-1]*bs+s[i]);
```

假如我们对于字符串 *abcd* 求好了它的前缀哈希值。

$$\begin{aligned}hs[1] &= a * base^0 \\hs[2] &= a * base^1 + b * base^0 \\hs[3] &= a * base^2 + b * base^1 + c * base^0 \\hs[4] &= a * base^3 + b * base^2 + c * base^1 + d * base^0\end{aligned}$$

那么, 假如我们要求里面的一个区段, 如 *bc* 的哈希值, 该怎么求呢?

观察可得,

$$\begin{aligned}
 hs[2,3] &= b * base^1 + c * base^0 \\
 &= a * base^2 + b * base^1 + c * base^0 - a * base^2 \\
 &= hs[3] - hs[1] * base^{3-1}
 \end{aligned}$$

写成通式就是

$$hs[l,r] = hs[r] - hs[l-1] * base^{r-l+1}$$

代码如下

```
ll calc(int l,int r) {return mo(hs[r]-hs[l-1]*p[r-l+1]);}
```

二维哈希

假如我们有一个 $m * n$ 大小的字符矩阵，那么如何求这个矩阵的哈希值呢？

这时我们设两个进制 p, q ，横坐标上进一位则乘 p ，纵坐标上进一位乘 q 。假设我们有字符矩阵

知乎

首发于
算法杂记

...

$\begin{bmatrix} c & d \end{bmatrix}$

收起

那么它的哈希值为

$$hs = a * p^1 * q^1 + b * p^0 * q^1 + c * p^1 * q^0 + d * p^0 * q^0$$

求哈希值的方法和二维前缀和类似。

```
void init(void)
{
    ps[0]=qs[0]=1;
    for (int i=1;i<=N;i++) ps[i]=mo(ps[i-1]*p);
    for (int i=1;i<=N;i++) qs[i]=mo(qs[i-1]*q);
    for (int i=1;i<=m;i++)
        for (int j=1;j<=n;j++)
            hs[i][j]=mo(s[i][j]+hs[i-1][j]*p);
    for (int i=1;i<=m;i++)
        for (int j=1;j<=n;j++)
            hs[i][j]=mo(hs[i][j]+hs[i][j-1]*q);
}
```

同理，如果我们要求一个子矩阵 (x_1, y_1, x_2, y_2) 的哈希值，那么操作也和二维前缀和类似。

$$\begin{aligned}
 hs(x_1, y_1, x_2, y_2) &= hs[x_2][y_2] - hs[x_1-1][y_2] * p^{x_2-x_1+1} - hs[x_2][y_1-1] \\
 &\quad * q^{y_2-y_1+1} + hs[x_1-1][y_1-1] * p^{x_2-x_1+1} * q^{y_2-y_1+1}
 \end{aligned}$$

以上便是字符串哈希的大致内容。

例题

【模板】字符串哈希 - 洛谷

www.luogu.com.cn/problem/P3370

模板题，不用多说了。

[BeiJing2011] Matrix 矩阵哈希

www.luogu.com.cn/problem/P10474

二维哈希模板题。

二维哈希

二维哈希



赞同 1



分享

```

#include<stdio.h>
#include<string.h>
#include<algorithm>
#include<map>
using namespace std;
typedef long long ll;
const int N=1e3;
const ll mod=1e9+7,p=131,q=13331;
map<ll,bool> vis;
int m,n,a,b,t;
ll ps[N+5],qs[N+5],hs[N+5][N+5];
char s[N+5][N+5];
ll mo(ll u) {return u;}
void init(void)
{
    ps[0]=qs[0]=1;
    for (int i=1;i<=N;i++) ps[i]=mo(ps[i-1]*p);
    for (int i=1;i<=N;i++) qs[i]=mo(qs[i-1]*q);
    for (int i=1;i<=m;i++)
        for (int j=1;j<=n;j++)
            hs[i][j]=mo(s[i][j]+hs[i-1][j]*p);
    for (int i=1;i<=m;i++)
        for (int j=1;j<=n;j++)
            hs[i][j]=mo(hs[i][j]+hs[i][j-1]*q);
    for (int i=a;i<=m;i++)
        for (int j=b;j<=n;j++)
        {
            vis[mo(hs[i][j]-hs[i-a][j]*ps[a]-hs[i][j-b]*qs[b]+hs[i-a][j-b]*ps[i-a]*qs[b])] = true;
            //printf("%lld\n",mo(hs[i][j]-hs[i-a][j]*ps[a]-hs[i][j-b]*qs[b]+hs[i-a][j-b]*ps[i-a]*qs[b]));
        }
}
int main(void)
{
    scanf("%d%d%d%d",&m,&n,&a,&b);
    for (int i=1;i<=m;i++) scanf("%s",s[i]+1);
    init();
    scanf("%d",&t);
    while (t--)
    {
        for (int i=1;i<=a;i++) scanf("%s",s[i]+1);
        for (int i=1;i<=a;i++)
            for (int j=1;j<=b;j++)
                hs[i][j]=mo(s[i][j]+hs[i-1][j]*p);
        for (int i=1;i<=a;i++)
            for (int j=1;j<=b;j++)
                hs[i][j]=mo(hs[i][j]+hs[i][j-1]*q);
        //printf("%lld\n",hs[a][b]);
        if (vis[hs[a][b]]) printf("1\n");
        else printf("0\n");
    }
    return 0;
}

```

[POI2012] PRE-Prefixuffix

www.luogu.com.cn/problem/P3546

一道哈希紫题，涉及到kmp算法。

```

#include<stdio.h>
#include<string.h>
#include<algorithm>
using namespace std;
typedef long long ll;
const int N=1e6;
const ll mod=1e9+7,bs=131;

```

```

int n,nxt[N+5],f[N+5],ans;
ll p[N+5],hs[N+5];
char s[N+5];
ll mo(ll u) {return ((u%mod+mod)%mod);}
ll calc(int l,int r) {return mo(hs[r]-hs[l-1]*p[r-l+1]);}
int main(void)
{
    scanf("%d%s",&n,s+1);
    for (int i=2,j=0;i<=n;i++)
    {
        while (j&&s[j+1]!=s[i]) j=nxt[j];
        if (s[j+1]==s[i]) j++;
        nxt[i]=j;
    }
    p[0]=1;
    for (int i=1;i<=n;i++) p[i]=mo(p[i-1]*bs);
    for (int i=1;i<=n;i++) hs[i]=mo(hs[i-1]*bs+s[i]);
    for (int i=n/2;i-->0)
    {
        f[i]=f[i+1]+2;
        while (f[i]&&(i+f[i])*2>n) f[i]--;
        while (f[i]&&calc(i+1,i+f[i])!=calc(n-f[i]-i+1,n-i)) f[i]--;
    }
    for (int i=nxt[n];i;i=nxt[i]) if (i*2<=n) ans=max(ans,i+f[i]);
    printf("%d\n",ans);
    return 0;
}

```

完结撒花~

编辑于 2024-07-30 17:16 · IP 属地江苏

内容所属专栏



算法杂记

分享关于C++算法的学习笔记与题解。

订阅专栏

算法

数据结构

OI (信息学奥林匹克)



理性发言，友善互动



还没有评论，发表第一个评论吧

推荐阅读