

# 字符串哈希

给定一个长度为  $n$  的字符串，再给定  $m$  个询问，每个询问包含四个整数  $l_1, r_1, l_2, r_2$ ，请你判断  $[l_1, r_1]$  和  $[l_2, r_2]$  这两个区间所包含的字符串子串是否完全相同。

字符串中只包含大小写英文字母和数字。

### 输入格式

第一行包含整数  $n$  和  $m$ ，表示字符串长度和询问次数。

第二行包含一个长度为  $n$  的字符串，字符串中只包含大小写英文字母和数字。

接下来  $m$  行，每行包含四个整数  $l_1, r_1, l_2, r_2$ ，表示一次询问所涉及的两个区间。

注意，字符串的位置从 1 开始编号。

### 输出格式

对于每个询问输出一个结果，如果两个字符串子串完全相同则输出 `Yes`，否则输出 `No`。

每个结果占一行。

### 数据范围

$$1 \leq n, m \leq 10^5$$

### 输入样例：

### 输入样例：

```
8 3
aabbbaabb
1 3 5 7
1 3 6 8
1 2 1 2
```

### 输出样例：

```
Yes
No
Yes
```

理解：

```
#include<iostream>
using namespace std;

typedef unsigned long long ULL;          //由于前缀值的值会很大 所以应该将数组中的数据定义为ULL型

const int N=1e5+10;
const int P=131;                        //P为权重
                                         //131为经验值 即P=131或13331时 哈希冲突的可能性最小

int n,m;

char str[N];
ULL h[N];                                //h[] 存放字符串的前缀值
ULL p[N];                                //p[] 存放各个位数的相应权值
```

```

ULL get(int l,int r)
{
    return h[r]-h[l-1]*p[r-l+1];    //这步其实是将h[l-1]左移
}                                    //其目的事实上是为了将h[l-1]的高位与h[r]相对齐
从而才可以未完成计算
int main()
{
    scanf("%d%d%s",&n,&m,str+1);

    p[0]=1;                          //注意这步千万不要忘了 最开始的权值必须赋值为1 否则接
    下来就会出错
    for(int i=1;i<=n;i++)
    {
        p[i]=p[i-1]*P;                //计算每个位上的相应权值
        h[i]=h[i-1]*P+str[i];         //计算字符串前缀值
        //最新加入的数的权值为p的0次 所以直接加上str[i]即可
    }
    while(m--)
    {
        int l1,r1,l2,r2;
        scanf("%d%d%d%d",&l1,&r1,&l2,&r2);
        if(get(l1,r1)==get(l2,r2)) puts("Yes");
        else puts("No");
    }

    return 0;
}

```

作者: Ni

链接: <https://www.acwing.com/solution/content/5553/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。

y总代码:

```

#include <iostream>
#include <algorithm>

using namespace std;

typedef unsigned long long ULL;

const int N = 100010, P = 131;

int n, m;
char str[N];
ULL h[N], p[N];

ULL get(int l, int r)
{
    return h[r] - h[l - 1] * p[r - l + 1];
}

int main()
{
    scanf("%d%d", &n, &m);

```

```

scanf("%s", str + 1);

p[0] = 1;
for (int i = 1; i <= n; i ++ )
{
    h[i] = h[i - 1] * P + str[i];
    p[i] = p[i - 1] * P;
}

while (m -- )
{
    int l1, r1, l2, r2;
    scanf("%d%d%d%d", &l1, &r1, &l2, &r2);

    if (get(l1, r1) == get(l2, r2)) puts("Yes");
    else puts("No");
}

return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/45313/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。