字符串哈希

String Hashing

Idea：将字符串视为 $base$ 进制的数，即：

$$hash_n = s_1 \cdot base^{n-1} + s_2 \cdot base^{n-2} + \cdots + s_{n-1} \cdot base + s_n$$

可选取大质数作为模数或 `unsigned long long` 自然溢出。

这种哈希方法的优点是可提取子串：

$$s_{l \ldots r} = hash_r - hash_{l-1} \cdot base^{r-l+1}$$

Code：

```
ULL h[LEN], base = 233; // hash value of string s[1...i] is stored in h[i]
void Hash(char s[]){
    int len = (int)strlen(s+1);
    h[1] = s[1];
    for(int i = 2; i <= len; i++)
        h[i] = h[i-1] * base + s[i];
}
ULL getSubstring(int l, int r){ // get hash value of s[l...r]
    return h[r] - h[l-1] * pow(base, r - l + 1);
}
```