字符串

前缀函数与 KMP

```
vector<size_t> prefix_function(const string& s)
{
    vector<size_t> nxt(s.length());
    for(size_t i=1,j=0;i<s.length();i++)
    {
        while(j>0&&s[j]!=s[i]) j = nxt[j-1];
        if(s[j]==s[i]) nxt[i] = ++j;
    }
    return nxt;
}
// split 为任意不在字符集中的字符
vector<size_t> kmp(const string& match, const string& pattern, char split = '#')
{
    vector<size_t> res;
    auto s = pattern+split+match;
    auto nxt = prefix_function(s);
    for(size_t i=0;i<s.length();i++)
        if(nxt[i]==pattern.length())
            res.push_back(i-(pattern.length()<<1));
    return res;
}
```