

【第三十周】经典匹配算法：KMP、Sunday 与 Shift-[AndOr] 算法

KMP、Sunday 与 Shift-[AndOr] JS版本

```
function getNext(pattern, next) {
    let n = pattern.length;
    next[0] = -1;
    for (let i = 1, j = -1; i < n; i++) {
        while (j != -1 && pattern[i] != pattern[j + 1]) j = next[j];
        if (pattern[i] == pattern[j + 1]) j += 1;
        next[i] = j;
    }
}

function KMP(text, pattern) {
    let n = text.length, m = pattern.length;
    let next = new Array(m);
    getNext(pattern, next);
    for (let i = 0, j = -1; i < n; i++) {
        while (j != -1 && text[i] != pattern[j + 1]) j = next[j];
        if (text[i] == pattern[j + 1]) j += 1;
        if (j + 1 == m) return i - j;
    }
    return -1;
}

function sunday(text, pattern) {
    let n = text.length, m, last_pos = new Array(256).fill(-1);
    for (m = 0; pattern[m]; ++m) last_pos[pattern[m].charCodeAt()] = m;
    for (let i = 0; i + m <= n; i += (m - last_pos[text[i + m].charCodeAt()]))
    {
        let flag = 1;
        for (let j = 0; pattern[j]; ++j) {
            if (text[i + j] == pattern[j]) continue;
            flag = 0;
            break;
        }
        if (flag) return i;
    }
    return -1
}
```

```
// console.log(last_pos)
}

function shift_and(text, pattern) {
    let n = pattern.length, p = 0;
    let code = new Array(256);
    for (let i = 0; i < n; i++) code[pattern[i]] |= 1 << i;
    for (let i = 0; text[i]; i++) {
        p = ((p << 1 | 1) & code[text[i]]);
        if (p & (1 << n - 1)) return i - n + 1;
    }
    return -1;
}

var s1 = "afsadfgasdgf";
var s2 = "dgf";
console.log(KMP(s1, s2));
console.log(shift_and(s1, s2));
console.log(sunday(s1, s2));
```



开课吧