

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

BZOJ4179 B

2019-01-07 | 题解

今天模拟赛的第二题，还是大水题，但是由于 SPFA 打错调了好久，还有这个 SB 出题人怎么说都不说一句读入的 L 是 long long 范围的啊，你™咋不上高精？

好了，回归正常的题解。本题可以理解为 POI2000 病毒 的加强版（连样例都一样），于是按照惯例，我们只需要建造一棵 AC 自动机。如果能够找到环，那么一定能够产生无限长度的答案串，直接输出 Yes 即可。否则就是一棵 DAG，跑一下 SPFA（或 BFS）求出最长的答案串，和要求的 L 比较一下就好。

~~我不会说我写了个 SPFA 还写挂调了两个小时的...~~

```
// =====  
//   author: memset0  
//   date: 2019.01.07 08:46:27  
//   website: https://memset0.cn/  
// =====  
#include <bits/stdc++.h>  
#define ll long long  
namespace ringo {  
template <class T> inline void read(T &x) {
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

```

x = 0; register char c = getchar(); register bool f = 0;
while (!isdigit(c)) f ^= c == '-', c = getchar();
while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10, M = 2e6 + 10;
int n, x, tot, q[M], dis[M];
bool inq[M], vis[M], found[M], find_answer;
char a[N];
ll m;

struct node {
    int ch[2], fail;
    bool tag;
} e[M];

void insert(char *a) {
    int len = strlen(a), p = 1;
    for (int i = 0, x; i < len; i++) {
        x = a[i] == 'A' ? 0 : 1;
        if (!e[p].ch[x]) e[p].ch[x] = ++tot;
        p = e[p].ch[x];
    }
    e[p].tag = 1;
}

void build_fail() {
    std::queue<int> q; int p, t;
    if (e[1].ch[0]) q.push(e[1].ch[0]), e[e[1].ch[0]].fail = 1;
    if (e[1].ch[1]) q.push(e[1].ch[1]), e[e[1].ch[1]].fail = 1;

```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memseto](#)
Proudly powered by [Typecho](#)

```
while (q.size()) {
    p = q.front(), q.pop();
    for (int i = 0; i < 2; i++)
        if (e[p].ch[i]) {
            q.push(e[p].ch[i]);
            t = e[p].fail;
            while (t != 1 && !e[t].ch[i]) t = e[t].fail;
            if (!e[t].ch[i]) e[e[t].ch[i]].fail = 1;
            else e[e[p].ch[i]].fail = e[t].ch[i], e[e[p].ch[i]].tag |= e[
        } else e[p].ch[i] = e[e[p].fail].ch[i];
    }
}

void find_circle(int u) {
    if (find_answer) return;
    vis[u] = 1;
    for (int i = 0; i < 2; i++) {
        if (!e[u].ch[i]) continue;
        if (vis[e[u].ch[i]]) {
            find_answer = true; return;
        } else if (!e[e[u].ch[i]].tag && !found[e[u].ch[i]]) {
            found[e[u].ch[i]] = 1;
            find_circle(e[u].ch[i]);
        }
    }
    vis[u] = 0;
}

bool spfa() {
    int u, v, l = 0, r = 0, max = 0; q[0] = 1, dis[1] = 0;
    while (l <= r) {
        u = q[(l++) % tot];
        for (int i = 0; i < 2; i++) {
            v = e[u].ch[i];
            if (dis[u] + 1 > dis[v] && v && !e[v].tag) {
                dis[v] = dis[u] + 1;
                if (!inq[v]) inq[q[(++r) % tot] = v];
            }
        }
    }
}
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

```
    }  
    }  
    }  
    for (int i = 1; i <= tot; i++) max = std::max(dis[i], max);  
    return max >= m;  
}  
  
void main() {  
    while (scanf("%d%lld", &n, &m) != EOF) {  
        for (int i = 1; i <= tot; i++) e[i].ch[0] = e[i].ch[1] = e[i].fail =  
            find_answer = 0, tot = 1;  
        for (int i = 1; i <= n; i++) scanf("%s", a), insert(a);  
        build_fail(), find_circle(1);  
        if (!find_answer) find_answer = spfa();  
        puts(find_answer ? "Yes" : "No");  
        for (int i = 1; i <= tot; i++) vis[i] = found[i] = dis[i] = inq[i] =  
            0;  
    }  
}  
  
} signed main() { return ringo::main(), 0; }
```

AC 自动机

SPFA

用户名

邮箱

网址 (选填)

可以在这里写评论哦 ~



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

提交评论

BZOJ5403 marshland
上一篇 «

BZOJ2410 Nim游戏
» 下一篇

© 2017 - 2019 **memset0** 的博客.
浙ICP备19006255号-1
97688 visits · 24756 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

