Xbwwb 递归

xbwxwbbwb

xwbbw

xxbwwbbbw

typedef long long ll;

//#define IO ios\_base::sync\_with\_stdio(0),cin.tie(0)

const int maxn = 10005;

vector<int> ans[maxn];

int q[maxn];string s;

int output(int p)

{

if( ans[p].size() == 0)

cout << s[p];

else

{

cout << 'x';

for(int i = 0; i < 4; i++)

{

int v = (i+2)%4;

// cout << p << ' ' << v << ' ' << ans[p][v] << endl;

/// 输出 第p层的第v个 (0<=v<4)

output( ans[p][v] );

// ans[p][2] ans[p][3] ans[p][0] ans[p][1]

}

}

}

int main()

{

int t;cin >> t;

while( t-- )

{

cin >> s; int len = s.size();

for(int i = 0; i < maxn; i++) ans[i].clear();

int top = 0;/// top :层数 递归的第几层 多深 深度

/// q数组: q[top] = i 无重复的层数 从第i个开始有分叉

for(int i = 0; i < len; i++)

{

int \_top = top-1;

if( top && ans[ q[\_top] ].size() == 4)

// 如果栈中有元素(已经在递归中 往里放元素)&&已经读满四个(已经放满)

top--, \_top--;/// 退栈 退出当前的递归层数

if( s[i] == 'x') // 是时候该连续的读四个了

{

if(top) // 如果栈中有元素

ans[q[\_top]].push\_back(i);// 连续q[\_top]读入四个

q[top] = i;// 标记 从第i个开始有分叉 在第top层等于i

top++; // 递归的层数加1

}

else if(top) // 如果栈中有元素 则在递归中

ans[q[\_top]].push\_back(i); // 将第i个元素加入ans的q[\_top]层中

//cout << \_top << ' ' << q[\_top] <<' ' << i << endl;

}

output(0);

cout << endl;

}

}