**[Aho Corasick](http://lightoj.com/volume_problemcategory.php?category=Aho%20Corasick), O(M+N)**

**Tested on LOJ 1427**

int c[N+2][26+2] , tot , fail[N+2], st[N+2], top, val[N+2], pos[N+2] ; //N=total pattern length

int new\_node ()

{

val[tot] = fail[tot] = 0 ;

memset(c[tot], 0, sizeof(c[tot]));

return tot ++ ;

}

void insert ( char \*s , int id )

{

int now = 0 ;

for ( ; \*s ; s ++ )

{

int k = \*s - 'a' ;

if ( !c[now][k] ) c[now][k] = new\_node () ;

now = c[now][k] ;

}

pos[id] = now ;

}

void getFail ()

{

queue<int> Q ;

int u = 0 , i , j ,e ;

for ( i = 0 ; i < 26 ; i ++ )

if ( c[u][i] ) Q.push ( c[u][i] ) ;

while ( !Q.empty () )

{

u = Q.front () , Q.pop () ;

for ( i = 0 ; i < 26 ; i ++ )

if ( c[u][i] )

{

e = c[u][i] , j = fail[u] ;

fail[e] = c[j][i];

st[++top] = e ;

Q.push ( e ) ;

}

else c[u][i] = c[fail[u]][i] ;

}

}

void solve ( char \*s , int n )

{

int now = 0 ;

for ( ; \*s ; s ++ )

{

int k = \*s - 'a' ;

now = c[now][k] ;

val[now] ++ ;

}

while ( top )

{

int p = st[top] ;

val[fail[p]] += val[p] ;

top -- ;

}

}

int main()

{

int tc,kk=1, n;

char pattern[K+2], text[M+2]; //K=total pattern, M=text length

cin>>tc;

while(tc--)

{

cin>>n;

scanf("%s", text);

tot = top = 0 ;

new\_node () ;

for(int i=1;i<=n;i++)

{

scanf("%s", pattern);

insert(pattern, i);

}

getFail();

solve(text, n);

cout<<"Case "<<kk++<<":"<<endl;

for (int i = 1 ; i <= n ; i ++ )

printf ( "%d\n" , val[pos[i]] ) ;

}

return 0;

}