Strings

**DETERMINAR DADO UN STRING SI EXISTE UN SUBSTRING CUADRADO**

//un substring es un cuadrado si se cumple por ejemplo: "abab" es cuadrado por tener "ab" consecutivo

//no importa en que lugar de la cadena aparezca, pero si tiene que ser consecutivo

bool rtest(char \*u, int m, char \*v, int n) {

for (int i = n, j, q = n; i >= 0; i = max(j, i / 2) - 1) {

for (j = i; j >= 0 && m - 1 - i + j >= 0 && u[m - 1 - i + j] == v[j]; --j);

if (j < 0)

return true;

if (i + j <= q) {

for (q = i + j + 1; q > i && v[q] == v[q - i - 1]; --q);

if (q <= i)

return true;}}

return false; }

bool rec(char \*s, char \*r, int n) {

int m = n / 2;

if (n <= 1)

return false;

if (rec(s, r + n - m, m) || rec(s + m, r, n - m))

return true;

if (rtest(s, m, s + m, n - m) || rtest(r, n - m, r + n - m, m))

return true;

return false; }

bool square(char \*s, int n) {

char r[n]; copy(s, s + n, r); reverse(r, r + n);return rec(s, r, n);}

**LONGEST INCREASING SUBSTRING**

const int inf = 99999999;

#define index distance(A.begin(), lower\_bound(A.begin(), A.end(), a[i]))

vector<int> lis\_fast(const vector<int> & a) {

const int n = a.size(); vector<int> A(n, inf); vector<int> id(n);

for (int i = 0; i < n; ++i) {

id[i] = index;

A[id[i]] = a[i];}

int m = \*max\_element(id.begin(), id.end());

vector<int> b(m + 1);

for (int i = n - 1; i >= 0; --i)if (id[i] == m) b[m--] = a[i];return b; }

**STRING MANAGER**

/\*Vector rad[] is really the length of the palindromes,

that is, rad[i] is the length of the palindrome centered

at i (odd palindromes are in rad[ 2 \* i ] and even ones

in rad[ 2 \* i + 1 ])\*/

#define MAXLEN 100000

int n, i, j, k;

int rad[ 2 \* MAXLEN ]; char s[MAXLEN];

int main() {

scanf( "%s", s );

n = strlen( s );

for ( i = 0, j = 0; i < 2 \* n - 1; i += k ) {

while ( i - j >= 0 && i + j + 1 < 2 \* n &&

s[ ( i - j ) / 2 ] == s[ ( i + j + 1 ) / 2 ] )

j++;

rad[i] = j;

for ( k = 1; k <= rad[i] && rad[ i - k ] != rad[i] - k; k++ )

rad[ i + k ] = min( rad[ i - k ], rad[i] - k );

j = max( j - k, 0 ); } }}