Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8166832	04:31:12	Dvaput	✓ Accepted	0.40 s	C++
	✓✓✓✓✓✓✓✓✓✓✓✓✓				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
dvaput.cpp	1635 bytes	caa59a7d75fce99c61b2938a64ac6110b7474408	download

Edit and resubmit this submission.

dvaput.cpp

```
//https://open.kattis.com/problems/dvaput
#include <bits/stdc++.h>
using namespace std;

typedef vector<int> vi;
typedef vector<vi> vvi;

// suffixarray
struct S { int l, r, p; };
bool operator<(const S &lhs, const S &rhs) {
    return lhs.l != rhs.l ? lhs.l < rhs.l : lhs.r < rhs.r;
}
bool operator==(const S &lhs, const S &rhs) {
    return lhs.l == rhs.l && lhs.r == rhs.r;
}

struct SuffixArray {
    string s;
    int n;
    vvi P;
    SuffixArray(string &_s) : s(_s), n(_s.length()) { construct(); }
    void construct() {
        vector<S> L(n, {0, 0, 0});
        P.push_back(vi(n, 0));
        for (int i = 0; i < n; ++i) P[0][i] = int(s[i]);
        for (int k = 1, cnt = 1; cnt / 2 < n; ++k, cnt *= 2) {
            P.push_back(vi(n, 0));
            for (int i = 0; i < n; ++i)
                L[i] = { P[k - 1][i], i + cnt < n
                    ? P[k - 1][i + cnt] : -1, i};
            sort(L.begin(), L.end());
            for (int i = 0; i < n; ++i)
                P[k][L[i].p] = (i > 0 && L[i] == L[i - 1]
                ? P[k][L[i - 1].p] : i);
        }
    }

    vi &get_array() { return P.back(); }
    int lcp(int x, int y) {
        int k, ret = 0;
        if (x == y) return n - x;
        for (int k = P.size() - 1; k >= 0 && x < n && y < n; --k)
            if (P[k][x] == P[k][y]) {
                x += 1 << k;
                y += 1 << k;
                ret += 1 << k;
            }
        return ret;
    }
};
// -suffixarray

int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);

    int L;
    string s;
    cin >> L >> s;
    SuffixArray sa(s);
    vector<int> inv(L, 0);
    for (int i = 0; i < L; ++i)
        inv[sa.get_array()[i]] = i;

    int M = 0;
    for (int i = 0; i < L - 1; ++i) {
        M = max(M, sa.lcp(inv[i], inv[i + 1]));
    }
    cout << M << endl;
    return 0;
}
```