

356e rng

```
using namespace std;
```

```
#define REP(i,n) for((i)=0;(i)<(int)(n);(i)++)
```

```
#define snuke(c,itr) for(__typeof((c).begin()) itr=(c).begin();itr!=  
=(c).end();itr++)
```

```
typedef long long ll;
```

```
int N;
```

```
string s;
```

```
#define X 6543653456536711
```

```
int freq[30][100010];
```

```
ll power[100010];
```

```
ll hash[100010];
```

```
void pre(void){
```

```
    int i,j;
```

```
    REP(i,26) REP(j,N) freq[i][j+1] = freq[i][j] + ((s[j] == 'a' + i) ?  
1 : 0);
```

```
    power[0] = 1;
```

```
    REP(i,N) power[i+1] = power[i] * X;
```

```
    REP(i,N) hash[i+1] = hash[i] + power[i] * (ll)(s[i] - 'a');
```

```
}
```

```
bool same(int p, int q, int len){  
    if(p > q) swap(p, q);  
    if(q + len > N) return false;  
    ll hash1 = (hash[p+len] - hash[p]) * power[q-p];  
    ll hash2 = hash[q+len] - hash[q];  
    return (hash1 == hash2);  
}
```

```
int common(int p, int q){  
    int ans=0,i;  
    for(i=16;i>=0;i--) if(same(p, q, ans+(1<<i))) ans += (1<<i);  
    return ans;  
}
```

```
int gray_len[20];  
bool gray[20][100010];  
ll cost[100010];  
  
ll init_score(void){  
    int i,j;  
  
    gray_len[0] = 1;  
    for(i=1;i<=15;i++) gray_len[i] = gray_len[i-1] * 2 + 1;
```

```

REP(i, 16) REP(j, N) if(j + gray_len[i] <= N) {
    if(i == 0) {
        gray[i][j] = true;
    } else {
        int p = j;
        int q = j + gray_len[i-1] + 1;
        if(!gray[i-1][p] || !gray[i-1][q] || !same(p, q,
gray_len[i-1])) continue;
        int c = s[q-1] - 'a';
        if(freq[c][j+gray_len[i]] - freq[c][j] != 1) continue;
        gray[i][j] = true;
    }
}

ll ans = 0;
REP(i, 16) REP(j, N) if(gray[i][j]) {
    ll tmp = (ll)gray_len[i] * gray_len[i];
    ans += tmp;
    cost[j] += tmp;
    cost[j+gray_len[i]] -= tmp;
}

REP(i, N) cost[i+1] += cost[i];

return ans;

```

```
}
```

```
11 score[100010][30];
```

```
void func(int level, int p){
```

```
    int i;
```

```
    if(level == 0){
```

```
        REP(i,26) score[p][i]++;
```

```
        return;
```

```
    }
```

```
    if(p + gray_len[level] > N) return;
```

```
    int d = gray_len[level-1];
```

```
    int q = p + d + 1;
```

```
    ll tmp = (ll)gray_len[level] * gray_len[level];
```

```
    // change center
```

```
    if(gray[level-1][p] && gray[level-1][q] && same(p, q, d)) REP(i,26)
```

```
    if(freq[i][q-1] - freq[i][p] == 0) score[q-1][i] += tmp;
```

```
    int len = common(p, q);
```

```
    if(len >= d) return;
```

```
    int p2 = p + len, q2 = q + len;
```

```
    if(common(p2 + 1, q2 + 1) + len + 1 < d) return;
```

```

    int center = s[q-1] - 'a';

    // change left

    if(gray[level-1][q] && freq[center][q+d] - freq[center][q-1] == 1)
score[p2][s[q2]-'a'] += tmp;

    // change right

    if(gray[level-1][p] && freq[center][q] - freq[center][p] == 1)
score[q2][s[p2]-'a'] += tmp;
}

char buf[100010];

int main(void){

    int i,j;

    scanf("%s", buf);

    s = buf;

    N = s.length();

    pre();

    ll ans = init_score();

    // cout << ans << endl;

    REP(i,16) REP(j,N) func(i, j);

```

```

    ll add = 0;

    REP(i,N) REP(j,26) if(s[i] != 'a' + j) add = max(add, score[i][j] -
cost[i]);

    cout << ans + add << endl;

    return 0;
}

```

tomasz

```

#define FOR(i, a, b) for(int i = (a); i <= (b); ++i)
#define FORD(i, a, b) for(int i = (a); i >= (b); --i)
#define REP(i, n) for(int i = 0; i < (n); ++i)
#define VAR(v, i) __typeof(i) v = (i)
#define FORE(i, c) for(VAR(i, (c).begin()); i != (c).end(); ++i)
#define ALL(x) (x).begin(), (x).end()
#define SZ(x) ((int)(x).size())
#define CLR(x) memset((x), 0, sizeof(x))
#define PB push_back
#define MP make_pair
#define X first
#define Y second
#define SQR(a) ((a) * (a))

```

```

#define DEBUG 1

#define debug(x) { if(DEBUG) cerr << #x << " = " << x << endl; }

#define debugv(x) { if (DEBUG) cerr << #x << " = "; FORE(iit, (x)) cerr
<< *iit << ", "; cout << endl; }


using namespace std;

typedef long long LL;

typedef long double LD;

typedef pair<int, int> P;

typedef vector<int> VI;

const int INF=((LL)1E9 + 7);

template<class C> void mini(C&aaaa, C bbbb) { aaaa = min(aaaa, bbbb); }

template<class C> void maxi(C&aaaa, C bbbb) { aaaa = max(aaaa, bbbb); }


string s;

struct gray {

    int pos, d;

    pair<int, char> change;

    char get(int i) const {

        int p = pos - (1<<d) + (1<<i);

        if (p == change.X) return change.Y;

        else return s[p];

    }

    gray(int _pos = -1, int _d = 0): pos(_pos), d(_d), change(MP(-1, 0))

```

```
{}
```

```
void print() {  
    cout << "[";  
    REP(i, d+1) cout << get(i);  
    cout << "]";  
    cout << change.X << change.Y << endl;  
}  
};  
  
void match (gray const & a, gray const & b, int pos, char mid,  
vector<gray> &res) {  
    int l = a.d;  
    FORD(i, l, 0) if (a.get(i) != b.get(i)) return;  
    if (a.change.X >= 0 && b.change.X >= 0) return;  
    bool prev[26];  
    REP(i, 26) prev[i] = false;  
    FOR(i, 0, l) prev[a.get(i)-'a']=true;  
    REP(i, 26) {  
        if (prev[i]) continue;  
        if (i != mid-'a') {  
            if (a.change.X == -1 && b.change.X == -1) {  
                gray g(pos, a.d+1);  
                g.change = MP(pos, i+'a');  
                res.PB(g);  
            }  
        }  
    }  
}
```



```

    } else {
        gray g(pos, a.d+1);
        if (a.change.X >= 0) g.change = a.change;
        else g.change = b.change;
        res.PB(g);
    }
}
}

```

```

const int MN = 100100;
const int LG = 18;
vector<gray> tab[2][MN];
LL delta[MN];
LL M[MN][26];

```

```

void init(int pos, char mid, vector<gray> &res) {
    REP(i, 26) {
        gray g(pos, 0);
        if (i != mid - 'a') {
            g.change = MP(pos, i+'a');
        }
        res.PB(g);
    }
}

int main() {

```

```

ios_base::sync_with_stdio(false);

cin >> s;

int n = SZ(s);

REP(i, n) init(i, s[i], tab[0][i]);

int d = 3;

int ph = 1;

LL sum = n;

while(d <= n) {

    LL val = (LL)d*d;

    REP(i, n-d+1) {

        tab[ph][i+d/2].clear();

        vector<gray> &left = tab[ph^1][i+d/4];

        vector<gray> &right = tab[ph^1][i+(3*d)/4];

        FORE(it, left) FORE(jt, right) {

            match(*it, *jt, i+d/2, s[i+d/2], tab[ph][i+d/2]);

        }

        FORE(it, tab[ph][i+d/2]) {

            if (it->change.X == -1) {

                delta[i] += val;

                delta[i+d] -= val;

                sum += val;

            } else M[it->change.X][it->change.Y-'a'] += (LL)d*d;

        }

    }

    d = 2*d+1;

```

```

        ph = (ph+1)%2;
    }

    LL bst = 0;

    LL cur = 0;

    REP(p, n) {

        cur += delta[p];

        REP(a, 26) maxi(bst, M[p][a]-cur);

    }

    cout << sum + bst << endl;

    return 0;

}

```