基础模板

快读快写

```cpp
#include<unordered_set>
#include<unordered_map>
#include<functional>
#include<algorithm>
#include<string.h>
#include<iostream>
#include<iterator>
#include<cstring>
#include<numeric>
#include<assert.h>
#include<cstdio>
#include<vector>
#include<bitset>
#include<queue>
#include<stack>
#include<cmath>
#include<set>
#include<map>
#define x first
#define y second
using namespace std;
//================================
#define FASTIO cin.tie(nullptr) -> sync_with_stdio(false)
#define debug(a) cout << #a": " << a << endl;
#define rep(i, ll, rr) for(int i = ll; i <= rr; ++ i)
#define per(i, rr, ll) for(int i = rr; i >= ll; -- i)
typedef pair<int,int> pii;
typedef long long LL;
typedef unsigned long long ULL;
typedef long double LD;
inline LL read() {
    LL s=0,w=1;
    char ch=getchar();
    for(; !isdigit(ch); ch = getchar())if(ch == '-') w = -1;
    for (; isdigit(ch); ch = getchar())s=(s<<1)+(s<<3)+(ch^48);
    return s*w;
}
inline void print(LL x,int op=10) {
    if(!x) {
        putchar('0');
        if(op)putchar(op);
        return;
    }
    char F[40];
    LL tmp=x>0?x:-x;
    if(x<0)putchar('-');
    int cnt=0;
    while(tmp>0) {
        F[cnt++]=tmp%10+'0';
        tmp/=10;
    }
    while(cnt>0)putchar(F[--cnt]);
    if(op)putchar(op);
}
inline void print128(__int128_t x) {
    if(x < 0) {
        putchar('-');
        x = -x;
    }
    if(x/10) print128(x/10);
    putchar(x%10+'0');
}
template <typename T>void read(T &x) {
    x=0;
    int f=1;
    char ch=getchar();
    while(!isdigit(ch)) {
        if(ch=='-')f=-1;
        ch=getchar();
    }
    while(isdigit(ch)) {
        x=x*10+(ch^48);
        ch=getchar();
    }
    x*=f;
    return;
}
template <typename T>void write(T x) {
    if(x<0) {
        putchar('-');
        x=-x;
    }
    if(x>9)write(x/10);
    putchar(x%10+'0');
    return;
}
LL fpower(LL a,LL b,LL mod) {
    LL ans = 1;
    while(b) {
        if(b & 1) ans = ans * (a % mod) % mod;
        a = a % mod * (a % mod) % mod;
        b >>= 1;
    }
    return ans;
}
LL Mod(LL a,LL mod) {
    return (a%mod+mod)%mod;
}
LL gcd(LL a,LL b) {
    return b?gcd(b,a%b):a;
}
int mov[8][2]= {1,0,0,1,-1,0,0,-1,1,1,-1,-1,1,-1,-1,1};
//==================================
// #define int LL
const int N=200010,M=N*2,mod=1e9+7;
```

```
int n,m,k,a[N];
string p;

void solve() {

}
//=================================
signed main() {
    int _=1;
    _=read();
    while(_--)  solve();
    return 0;
}
```

带几何的快读快写

```
#include<unordered_set>
#include<unordered_map>
#include<functional>
#include<algorithm>
#include<string.h>
#include<iostream>
#include<iterator>
#include<cstring>
#include<numeric>
#include<cstdio>
#include<vector>
#include<bitset>
#include<queue>
#include<stack>
#include<cmath>
#include<ctime>
#include<set>
#include<map>
#define x first
#define y second
using namespace std;
//================================DEBUG
#define FASTIO cin.tie(nullptr) -> sync_with_stdio(false)
#define debug(a) cout << #a": " << a << endl;
#define rep(i, ll, rr) for(int i = ll; i <= rr; ++ i)
#define per(i, rr, ll) for(int i = rr; i >= ll; -- i)
//================================IO
typedef long long LL; typedef unsigned long long ULL; typedef long double LD;
inline LL read(){LL s=0,w=1;char ch=getchar();for(;!isdigit(ch); ch = getchar())if(ch == '-') w = -1; for (; isdigit(ch);ch = getchar())s=(s<<1)+(s<<3)+(ch^48);return s*w;}
inline void print(LL x,int op=10){if(!x){putchar('0');if(op)putchar(op);return;}char F[40];LL tmp=x>0?x:-x;if(x<0)putchar('-');int cnt=0;while(tmp>0){F[cnt++]=tmp%10+'0';tmp/=10;}while(cnt>0)putchar(F[--cnt]);if(op)putchar(op);}
inline void print128(__int128_t x){if(x < 0) {putchar('-');x = -x;}if(x/10) print128(x/10);putchar(x%10+'0');}
template <typename T>void read(T &x){x=0;int f=1;char ch=getchar();while(!isdigit(ch)){if(ch=='-')f=-1;ch=getchar();}while(isdigit(ch)){x=x*10+(ch^48);ch=getchar();}x*=f;return;}
template <typename T>void write(T x){if(x<0){putchar('-');x=-x;}if(x>9)write(x/10);putchar(x%10+'0');return;}
//================================HABIT
LL fpower(LL a,LL b,LL mod) {LL ans = 1; while(b){ if(b & 1) ans = ans * (a % mod) % mod; a = a % mod * (a % mod) % mod; b >>= 1;} return ans; }
LL Mod(LL a,LL mod){return (a%mod+mod)%mod;}
LL gcd(LL a,LL b) {return b?gcd(b,a%b):a;}
int mov[8][2]={1,0,0,1,-1,0,0,-1,1,1,-1,-1,1,-1,-1,1};
//================================DEFINE
// #define int LL
// #define double long double
typedef pair<int,int> pii;
typedef pair<double,double> pdd;
const double eps=1e-9,PI=acos(-1),inf=1e9;
//================================GEOMETRY
int sign(double x) {if(fabs(x)<eps) return 0;return x>0?1:-1;}
struct Poi{
    double x,y;
    Poi operator-(Poi b){return {x-b.x,y-b.y};}
    Poi operator+(Poi b){return {x+b.x,y+b.y};}
    Poi operator*(double k){return {x*k,y*k};}
    Poi operator/(double k){return {x/k,y/k};}
    Poi norm(){double len=sqrt(x*x+y*y);return {x/len,y/len};}
    double operator*(Poi b){return x*b.y-y*b.x;}
    double operator&(Poi b){return x*b.x+y*b.y;}
    bool operator==(Poi b){return sign(x-b.x)==0&&sign(y-b.y)==0;}
    bool operator<(Poi b){return sign(x-b.x)<0||(sign(x-b.x)==0&&sign(y-b.y)<0);}
};
double cross(Poi a,Poi b){return a.x*b.y-a.y*b.x;}
double area(Poi a,Poi b,Poi c){return cross({b.x-a.x,b.y-a.y},{c.x-a.x,c.y-a.y});}
double dist(Poi a,Poi b){double dx=a.x-b.x;double dy=a.y-b.y;return sqrt(dx*dx+dy*dy);}
struct Cir{Poi p;double r;};
struct Line{Poi st,ed;};
//获得直线的角度
double get_angle(const Line &a){ return atan2(a.ed.y-a.st.y,a.ed.x-a.st.x);}
//直线按照角度的排序函数
bool cmp(Line &a,Line &b){  double A=get_angle(a),B=get_angle(b); if(sign(A-B)==0) return sign(area(a.st,a.ed,b.ed))<0;return A<B;}
//求直线p+kv和直线q+kw的交点
Poi get_line_intersection(Poi p,Poi v,Poi q,Poi w){ auto u=p-q; double t=cross(w,u)/cross(v,w); return {p.x+v.x*t,p.y+v.y*t};}
//两条线的交点
Poi get_line_intersection(Line a,Line b){ return get_line_intersection(a.st,a.ed-a.st,b.st,b.ed-b.st);}
//bc的交点是否再a的右侧
bool on_right(Line a,Line b,Line c){ auto jiao=get_line_intersection(b,c);return sign(area(a.st,a.ed,jiao))<=0;}
//将一个点顺时针旋转d度
Poi rotate(Poi a,double b){return {a.x*cos(b)+a.y*sin(b), -a.x*sin(b)+a.y*cos(b)};}
//获取中垂线
Line get_perpendicular_bisector(Poi a,Poi b){return {(a+b)/2,rotate(b-a,PI/2.0)};}
//三点确定圆
Cir get_cir(Poi a,Poi b,Poi c){auto u=get_perpendicular_bisector(a,b),v=get_perpendicular_bisector(a,c);auto p=get_line_intersection(u.st,u.ed,v.st,v.ed);return {p, dist(p,a)};}
double len(Poi a){return sqrt(a&a);}
bool on_segment(Poi p,Poi a,Poi b){ return !sign((p-a)*(p-b)) && sign((p-a)&(p-b))<=0;}//判断c是否在线段ab上
```

```
vector<Poi> get_circle_line_intersection(Poi a,Poi b,Cir c){ //线段ab和圆c的交点
    vector<Poi> ans;
    auto e=get_line_intersection(a, b-a, c.p, rotate(b-a,PI/2)); //弦与中垂线的交点
    auto d=dist(c.p, e); //弦心距
    if(!on_segment(e,a,b)) d=min(dist(c.p,a), dist(c.p, b));
    if(sign(c.r-d)<=0) return ans;
    auto len=sqrt(c.r*c.r-dist(c.p, e)*dist(c.p, e));
    Poi pa=e+(a-b).norm()*len,pb=e+((b-a).norm()*len);
    ans.push_back(pa);
    ans.push_back(pb);
    return ans;
}
double poi_to_segment(Poi a,Poi b,Poi c={0,0}){ //点到线段的距离
    auto e=get_line_intersection(a, b-a, c, rotate(b-a,PI/2)); //弦与中垂线的交点
    auto d=dist(c, e); //弦心距
    if(!on_segment(e,a,b)) d=min(dist(c,a), dist(c, b));
    return d;
}
//极角序
bool angle_cmp(const Poi &A,const Poi &B){
    auto quad=[](const Poi &A){if(A.y<-eps) return 1;if(A.y>eps) return 4;if(A.x<-eps) return 5;if(A.x>eps) return 3;return 2;};
    const int qa=quad(A),qb=quad(B);if(qa!=qb) return qa<qb;
    const auto t=A.x*B.y-B.x*A.y; //if(abs(t)<=eps) return (A&A)<(B&B)-eps;/*按极径排序*/
    return t>eps;
}
//c为圆，acb扇形面积
double sector_area(Poi a,Poi b,Cir c){ auto angle=acos((a&b)/len(a)/len(b));if(sign(a*b)<0) angle=-angle;return c.r*c.r*angle/2.0;}
/*random_suffle(p+1,p+1+n);  点随机化*/

/* 3D-GEOMETRY
double rand_eps(){return ((double)rand()/RAND_MAX-0.5)*eps;}
struct Point{ //三维点 or 向量
    double x,y,z;
    void shake(){x+=rand_eps(),y+=rand_eps(),z+=rand_eps();}
    Point operator+(Point b){return {x+b.x,y+b.y,z+b.z};}
    Point operator-(Point b){return {x-b.x,y-b.y,z-b.z};}
    double operator&(Point t){return x*t.x+y*t.y+z*t.z;} //点积
    Point operator*(Point t){return {y*t.z-t.y*z, z*t.x-x*t.z, x*t.y-y*t.x};}
    double len(){return sqrt(x*x+y*y+z*z);}
};
struct Plane{ //平面
    int v[3];
    Point norm(){return (q[v[1]]-q[v[0]])*(q[v[2]]-q[v[0]]);}
    double area() { return norm().len() / 2;}
    bool above(Point a){return ((a-q[v[0]])&norm()) >=0 ;}
    double dist(Point W){return (norm()&(q[v[0]]-W))/(norm().len());} //点到平面的距离
};*/
//=================================
const int N=200010,M=N*2,mod=1e9+7,INF=1e9;
int n,m,k,a[N];
string p;

void solve(){

}
//================================
signed main(){
    int _=1;
    // _=read();
    while(_--)  solve();
    return 0;
}
```

Py

```
import os,sys
import random
from io import BytesIO, IOBase

from collections import defaultdict,deque,Counter
from bisect import bisect_left,bisect_right
from heapq import heappush,heappop
from functools import lru_cache
from itertools import accumulate
import math

# Fast IO Region
BUFSIZE = 8192
class FastIO(IOBase):
    newlines = 0
    def __init__(self, file):
        self._fd = file.fileno()
        self.buffer = BytesIO()
        self.writable = "x" in file.mode or "r" not in file.mode
        self.write = self.buffer.write if self.writable else None
    def read(self):
        while True:
            b = os.read(self._fd, max(os.fstat(self._fd).st_size, BUFSIZE))
            if not b:
                break
            ptr = self.buffer.tell()
            self.buffer.seek(0, 2), self.buffer.write(b), self.buffer.seek(ptr)
        self.newlines = 0
        return self.buffer.read()
    def readline(self):
        while self.newlines == 0:
            b = os.read(self._fd, max(os.fstat(self._fd).st_size, BUFSIZE))
            self.newlines = b.count(b"\n") + (not b)
            ptr = self.buffer.tell()
            self.buffer.seek(0, 2), self.buffer.write(b), self.buffer.seek(ptr)
        self.newlines -= 1
        return self.buffer.readline()
    def flush(self):
        if self.writable:
            os.write(self._fd, self.buffer.getvalue())
            self.buffer.truncate(0), self.buffer.seek(0)
class IOWrapper(IOBase):
    def __init__(self, file):
        self.buffer = FastIO(file)
        self.flush = self.buffer.flush
        self.writable = self.buffer.writable
        self.write = lambda s: self.buffer.write(s.encode("ascii"))
        self.read = lambda: self.buffer.read().decode("ascii")
        self.readline = lambda: self.buffer.readline().decode("ascii")
sys.stdin, sys.stdout = IOWrapper(sys.stdin), IOWrapper(sys.stdout)
input = lambda: sys.stdin.readline().rstrip("\r\n")

def expand(s, left, right):
    while left >= 0 and right < len(s) and (s[left] == s[right] == '#' or s[left] + s[right] == '01' or s[left] + s[right] == '10'):
        left -= 1
        right += 1
    return (right - left - 2) // 2

def manachar(s):
    res = [0] * len(s)
    res2 = [0] * len(s)
    end, start = -1, 0
    s = '#' + '#'.join(list(s)) + '#'
    arm_len = []
    right = -1
    j = -1
    ans = 0
    for i in range(len(s)):
        if right >= i:
            i_sym = 2 * j - i
            min_arm_len = min(arm_len[i_sym], right - i)
            cur_arm_len = expand(s, i - min_arm_len, i + min_arm_len)
        else:
            cur_arm_len = expand(s, i, i)
        arm_len.append(cur_arm_len)
        if i + cur_arm_len > right:
            j = i
            right = i + cur_arm_len
        if 2 * cur_arm_len + 1 > end - start:
            start = i - cur_arm_len
            end = i + cur_arm_len
        if i % 2 == 0:
            if i // 2 < len(res2):
                ans += cur_arm_len
    return ans

# n, q = list(map(int, input().split(' ')))
# s = input()
# res1, res2 = manachar(s)
# print(res1, res2)
# pre1 = list(accumulate(res1))
# pre2 = list(accumulate(res2))
# for _ in range(q):
#     L, R = list(map(int, input().split(' ')))

# n, k = list(map(int, input().split(' ')))
# s = input()
# cnt = [0] * 1000005
# res1, res2 = manachar(s)
# for i in res1:
#     cnt[i] += 1
# for i in range(999999, 0, -2):
#     cnt[i] += cnt[i + 2]
# if sum(cnt) < k:
#     print(-1)
# else:
#     i = 999999
#     ans = 1
#     mod = 19930726
#     while i >= 1 and k > 0:
#         ans *= pow(i, min(k, cnt[i]), mod)
#         ans %= mod
#         k -= min(k, cnt[i])
#         i -= 2
#     print(ans)

# s = input()
# n = len(s)
# L = [1] * n
# R = [1] * n
# res1, res2 = manachar(s)
# for i in range(n):
#     L[i - res2[i] // 2] = max(L[i - res2[i] // 2],res2[i])
#     R[i + res2[i] // 2 - 1] = max(R[i + res2[i] // 2 - 1], res2[i])
#     L[i - res1[i] // 2] = max(L[i - res1[i] // 2], res1[i])
#     R[i + res1[i] // 2] = max(R[i + res1[i] // 2], res1[i])
# for i in range(1, n):
#     L[i] = max(L[i], L[i - 1] - 2)
# for i in range(n - 1)[::-1]:
#     R[i] = max(R[i], R[i + 1] - 2)
# ans = 0
# for i in range(n - 1):
#     ans = max(ans, R[i] + L[i + 1])
# print(ans)

# def expand(s, t, left, right):
#     while left >= 0 and right < len(t) and s[left] == t[right]:
#         left -= 1
#         right += 1
#     return (right - left - 2) // 2

# def manachar(s, t):
#     res = [0] * len(s)
#     res2 = [0] * len(s)
#     end, start = -1, 0
#     s = '#' + '#'.join(list(s)) + '#'
#     t = '#' + '#'.join(list(t)) + '#'
#     arm_len = []
#     right = -1
#     j = -1
#     for i in range(len(s)):
#         if right >= i:
#             i_sym = 2 * j - i
#             min_arm_len = min(arm_len[i_sym], right - i)
#             cur_arm_len = expand(s, t, i - min_arm_len, i + min_arm_len)
#         else:
#             cur_arm_len = expand(s, t, i, i)
#         arm_len.append(cur_arm_len)
#         if i + cur_arm_len > right:
```

```
#      j = i
#      right = i + cur_arm_len
#    if 2 * cur_arm_len + 1 > end - start:
#      start = i - cur_arm_len
#      end = i + cur_arm_len
#    if i % 2 == 1:
#      res[i // 2] = cur_arm_len
#    else:
#      if i // 2 < len(res2):
#        res2[i // 2] = cur_arm_len
#  return res, res2
# from random import randint
# import time
# t1 = time.time()
# s = ''.join([str(randint(0, 1)) for _ in range(500000)])
n = int(input())
s = input()
# t = ''
# t2 = time.time()
# for i in range(len(s)):
#  if s[i] == '1':
#    t += '0'
#  else:
#    t += '1'

# res1, res2 = manachar(s, t)
# res1, res2 = manachar(s)
print(manachar(s) // 2)
# print(sum(res2) // 2)
# t3 = time.time()
# print(t2-t1)
```