The event of zero and one
ACM Template

目录

1 String

1.1 Manacher

```
#include<bits/stdc++.h>
using namespace std;
const int MAX = 2e5+10000;
char s[MAX];
struct Manacher{
    int lc[MAX];
    char ch[MAX];
    int N;
    Manacher(char *s){init(s);manacher();}
    /* s 1 bas */
    void init(char *s){
        int n = strlen(s+1);
        ch[n*2 +1] = '#';
        ch[0] = '@';
        ch[n*2 +2] = '\0';
        for (int i=n;i>=1;i--){
            ch[i*2] = s[i];ch[i*2 -1] = '#';
        }
        N = 2* n +1;
    }
    void manacher(){
        lc[1]=1;  int k=1;
        for (int i=2;i<=N;i++){
            int p = k+lc[k]-1;
            if (i<=p){
                lc[i]=min(lc[2*k-i],p-i+1);
            }else{  lc[i]=1;  }
            while (ch[i+lc[i]]==ch[i-lc[i]])lc[i]++;
            if (i+lc[i]>k+lc[k])k=i;
        }
    }
    void debug(){
        puts(ch);
        for (int i=1;i<=N;i++){
            printf("lc[%d]=%d\n",i,lc[i]);
        }
    }
};
int main(){
    scanf("%s",s+1);
    Manacher manacher(s);
    manacher.debug();
    return 0;
}
```

2 Math

2.1 Linear$_s$*ieve*

```
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e7+10;
typedef long long ll;
bool used[maxn];
int mu[maxn];
vector<int> prime;
ll f[maxn];
int low[maxn];
void sieve(int size){
    //f:multiplicative function;
    assert(size < maxn);
    mu[1] = 1;
    f[1] = 1;
    for (int i=2;i<=size;i++){
        if (!used[i]){
            prime.push_back(i);
            mu[i] = -1;
            //f:TODO
            low[i] = i;
        }
        for (int j = 0;j < prime.size();j++){
            ll nxt = 1ll * i * prime[j];
            if (nxt > size)break;
            used[nxt] = 1;
            if (i % prime[j]){
                low[nxt] = prime[j];
                mu[nxt] = -mu[i];
                //f: mod or not?
                f[nxt] = f[i] * f[prime[j]];
            }else{
                low[nxt] = prime[j] * low[i];
```

```
8ec3                mu[nxt] = 0;
b401                if (low[nxt] != nxt){
427e                    //mod or not?
4d18                    f[nxt] = 1ll * f[low[nxt]] * f[nxt/low[nxt]];
8e2e                }else{
427e                    // i = prime[j] ^ k
427e                    //f:TODO
95cf                }
6173                break;
95cf            }
95cf        }
95cf    }
95cf}
3117int main(){
ff91    sieve(1e7);
7021    return 0;
95cf}
```