# Contents

# 1 Basic

## 1.1 default code

```
#include <bits/stdc++.h>
#define PB push_back
#define MP make_pair
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define ALL(x) (x).begin(),(x).end()
#ifdef _DEBUG_
  #define debug(...) printf(__VA_ARGS__)
#else
  #define debug(...) 0
#endif
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

int main() {

    return 0;
}
```

## 1.2 .vimrc

```
color torte
syn on
set guifont=Consolas:h16:
set number
set showcmd

” use indentation of previous line
set autoindent
” use intelligent indentation for C
set smartindent
” configure tabwidth and insert spaces instead of tabs
set tabstop=4        ” tab width is 4 spaces
set expandtab        ” expand tabs to spaces
set showmatch
” intelligent comments
set comments=sl:/*,mb:\ *,elx:\ */
set backspace=indent,eol,start
set softtabstop=4
set shiftwidth=4

map <F9> <ESC>:w<CR>:!g++ % -o %< -O2 -std=c++0x<CR>
map <S-F9> <ESC>:w<CR>:!g++ % -o %< -O2 -D_DEBUG_ -std=c++0x<CR>
map <F5> <ESC>:!./%<<CR>
map <F6> <ESC>:w<CR>ggvG”+y
map <S-F5> <ESC>:!./%< < %<.in<CR>
imap <Home> <ESC>^i
com INPUT sp %<.in
```

# 2 math

## 2.1 ext gcd

```
// find one solution  (x,y)  of  ax+by=gcd(a,b)
void ext_gcd(int a,int b,int &g,int &x,int &y)
{
  if(!b){ g=a; x=1; y=0; }
  else{ ext_gcd(b, a%b, g, y, x); y -= x*(a/b); }
}
```

# 3 flow

## 3.1 dinic

```
#include <bits/stdc++.h>
#define PB push_back
#define MP make_pair
#define F first
#define S second
#define SZ(x) ((int)(x).size())
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

/********************************/
// dinic
const int MAXV=300;
const int MAXE=10000;
const int INF=(int)1e9+10;

struct E{
  int to,co;//capacity
  E(int t=0,int c=0):to(t),co(c){}
}eg[2*MAXE];

// source:0  sink:n-1
struct Flow{
  VI e[MAXV];
  int ei,v;
  void init(int n) {
    v=n;
    ei=0;
    for(int i=0;i<n;i++)
      e[i]=VI();
  }
  void add(int a,int b,int c) { //a to b ,maxflow=c
    eg[ei]=E(b,c);
    e[a].PB(ei);
    ei++;
    eg[ei]=E(a,0);
    e[b].PB(ei);
    ei++;
  }

  int d[MAXV],qu[MAXV],ql,qr;
  bool BFS() {
    memset(d,-1,v*sizeof(int));
    ql=qr=0;
    qu[qr++]=0;
    d[0]=0;
    while(ql<qr && d[v-1]==-1) {
      int n=qu[ql++];
      VI &v=e[n];
      for(int i=v.size()-1;i>=0;i--) {
        int u=v[i];
        if(d[eg[u].to]==-1 && eg[u].co>0) {
          d[eg[u].to]=d[n]+1;
          qu[qr++]=eg[u].to;
        }
      }
    }
    return d[v-1]!=-1;
  }
  int ptr[MAXV];
  int go(int n,int p) {
    if(n==v-1)
      return p;
    VI &u=e[n];
    int temp;
    for(int i=ptr[n];i<SZ(u);i++)
    {
      if(d[n]+1!=d[eg[u[i]].to] || eg[u[i]].co==0)
        continue;
      if((temp=go(eg[u[i]].to,min(p,eg[u[i]].co)))==0)
        continue;
      eg[u[i]].co-=temp;
      eg[u[i]^1].co+=temp;
      ptr[n]=i;
      return temp;
    }
    ptr[n]=SZ(u);
    return 0;
  }
  int max_flow() {
    int ans=0,temp;
    while(BFS()) {
      for(int i=0;i<v;i++)
        ptr[i]=0;
      while((temp=go(0,INF))>0)
        ans+=temp;
    }
    return ans;
  }
}flow;

int main() {

  return 0;
}
```

# 4 string

## 4.1 KMP

```
/***
Test OJ 265
trivial string matching

input:
abc
abccbabbabc

output:
0 8

***/
```

```
#include <bits/stdc++.h>
#define PB push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

char S[500010],T[500010];
int K[500010];

int main()
{
  gets(S);
  gets(T);
  K[0]=-1;
  int a=-1;
  for(int i=1;S[i];i++)
  {
    while(a!=-1 && S[a+1]!=S[i])
      a=K[a];
    if(S[a+1]==S[i])
      a++;
    K[i]=a;
  }
  VI ans;
  a=-1;
  for(int i=0;T[i];i++)
  {
    while(a!=-1 && S[a+1]!=T[i])
      a=K[a];
    if(S[a+1]==T[i])
      a++;
    if(!S[a+1])
    {
      ans.PB(i-a);
      a=K[a];
    }
  }
  bool first=1;
  for(int u:ans)
  {
    if(first)
      printf(”%d”,u),first=0;
    else
      printf(” %d”,u);
  }
  puts(””);
  return 0;
}
```

## 4.2 Z-value

```
/***
Test OJ 265
trivial string matching

input:
abc
abccbabbabc
```

```

output:
0 8

***/
#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;

char S[1000010];
int Z[1000010];

int main()
{
  int len=0,lenS;
  gets(S);
  for(;S[len];len++);
  lenS=len;
  gets(S+len+1);
  for(len++;S[len];len++);
  S[len]=’*’;
  int bst=0;
  Z[0]=0;
  for(int i=1;i<len;i++)
  {
    if(Z[bst]+bst<i) Z[i]=0;
    else Z[i]=min(Z[bst]+bst-i,Z[i-bst]);
    while(S[Z[i]]==S[i+Z[i]]) Z[i]++;
    if(Z[i]+i>Z[bst]+bst) bst=i;
  }
  bool first=1;
  for(int i=lenS+1;i<len;i++)
    if(Z[i]>=lenS)
    {
      if(first)
        printf(”%d”,i-lenS-1),first=0;
      else
        printf(” %d”,i-lenS-1);
    }
  puts(””);
  return 0;
}
```

## 4.3 Suffix Array(N log N)

```
// NTUJ448
#include <bits/stdc++.h>
#define pb push_back
#define F first
#define S second
#define SZ(x) ((int)(x).size())
#define MP make_pair
using namespace std;
typedef long long ll;
typedef pair<int,int> PII;
typedef vector<int> VI;
```

```

const int SASIZE=2500000;
char in[500];
int S[SASIZE],from[SASIZE];
int R[SASIZE],SA[SASIZE],H[SASIZE];
int tR[SASIZE],tSA[SASIZE];
int cnt[SASIZE];
int num[4010];

int main()
{
  int N;
  while(scanf(”%d”,&N)==1 && N)
  {
    int len=0,maxR=0;
    for(int i=0;i<N;i++)
    {
      scanf(”%s”,in);
      for(int j=0;in[j];j++)
      {
        from[len]=i;
        S[len++]=in[j]-’a’;
      }
      from[len]=N;
      S[len++]=i+50;
    }
    memset(R,-1,sizeof(R));
    memset(cnt,0,sizeof(cnt));
    for(int i=0;i<len;i++)
    {
      R[i]=S[i];
      maxR=max(maxR,R[i]);
    }
    for(int i=0;i<len;i++)
      cnt[R[i]+1]++;
    for(int i=1;i<=maxR;i++)
      cnt[i]+=cnt[i-1];
    for(int i=0;i<len;i++)
      SA[cnt[R[i]]++]=i;
  /*  for(int i=0;i<len;i++)
      printf(”R[%d]=%d, SA[%d]=%d\n”,i,R[i],i,SA[i]);*/
    for(int i=1;i<len;i*=2)
    {
      memset(cnt,0,sizeof(int)*(maxR+10));
      memcpy(tSA,SA,sizeof(int)*(len+10));
      memcpy(tR,R,sizeof(int)*(len+i+10));
      for(int j=0;j<len;j++)
        cnt[R[j]+1]++;
      for(int j=1;j<=maxR;j++)
        cnt[j]+=cnt[j-1];
      for(int j=len-i;j<len;j++)
        SA[cnt[R[j]]++]=j;
      for(int j=0;j<len;j++)
      {
        int k=tSA[j]-i;
        if(k<0)
          continue;
        SA[cnt[R[k]]++]=k;
      }
      int num=0;
      maxR=0;
      R[SA[0]]=num;
      for(int j=1;j<len;j++)
      {
        if(tR[SA[j-1]]<tR[SA[j]] || tR[SA[j-1]+i]<tR[SA[j]+i])
```

```
          num++;
        R[SA[j]]=num;
        maxR=max(maxR,R[SA[j]]);
      }
    /*  puts(”-------”);
      for(int i=0;i<len;i++)
        printf(”R[%d]=%d, SA[%d]=%d\n”,i,R[i],i,SA[i]);*/
    }
    memset(H,0,sizeof(H));
    for(int i=0;i<len;i++)
    {
      if(R[i]==0)
        continue;
      int &t=H[R[i]];
      if(i>0)
        t=max(0,H[R[i-1]]-1);
      while(S[i+t]==S[SA[R[i]-1]+t]) t++;
    }
    /*for(int i=0;i<len;i++)
      printf(”R[%d]=%d, SA[%d]=%d\n”,i,R[i],i,SA[i]);
    for(int i=0;i<len;i++)
      printf(”%3d %3d  %s\n”,H[i],SA[i],S+SA[i]);*/
    /*for(int i=0;i<len;i++)
    {
      printf(”%3d %3d %d|”,H[i],SA[i],from[i]);
      for(int j=SA[i];j<len;j++)
        printf(”%2d ”,S[j]);
      puts(””);
    }*/
    memset(num,0,sizeof(num));
    int anslen=0,ansfrom=-1;
    int get=0;
    deque<PII> deq;
  /*  for(int i=0;i<len;i++)
      printf(”%d:%d\n”,i,from[i]);*/
    for(int l=0,r=0;r<len;r++)
    {
      if(from[SA[r]]<N && num[from[SA[r]]]==0)
        get++;
      num[from[SA[r]]]++;
      while(deq.size()>0 && deq.back().F>=H[r]) deq.pop_back();
      deq.pb(MP(H[r],r));
      while(num[from[SA[l]]]>1)
      {
        num[from[SA[l]]]--;
        l++;
      }
      while(deq.size()>0 && deq.front().S<=l) deq.pop_front();
      if(get==N && deq.front().F>anslen)
        anslen=deq.front().F, ansfrom=SA[l];
    }
    //printf(”(%d)\n”,anslen);
    if(anslen==0)
      puts(”IDENTITY LOST”);
    else
    {
      for(int i=ansfrom;i<ansfrom+anslen;i++)
        putchar(S[i]+’a’);
      puts(””);
    }
  }
  return 0;
}
```

# 5 graph

# 6 data structure

# 7 geometry