

Contents

1	Basic	2
1.1	default code	2
1.2	.vimrc	2
2	math	2
2.1	ext gcd	2
3	flow	3
3.1	dinic	3
4	string	4
4.1	KMP	4
4.2	Z-value	5
4.3	Suffix Array($N \log N$)	6
5	graph	9
6	data structure	9
7	geometry	9

1 Basic

1.1 default code

```

1 #include <bits/stdc++.h>
2 #define PB push_back
3 #define MP make_pair
4 #define F first
5 #define S second
6 #define SZ(x) ((int)(x).size())
7 #define ALL(x) (x).begin(),(x).end()
8 #ifdef _DEBUG_
9     #define debug(...) printf(__VA_ARGS__)
10 #else
11     #define debug(...) 0
12 #endif
13 using namespace std;
14 typedef long long ll;
15 typedef pair<int,int> PII;
16 typedef vector<int> VI;
17
18 int main() {
19
20     return 0;
21 }

```

1.2 .vimrc

```

1 color torte
2 syn on
3 set guifont=Consolas:h16:
4 set number
5 set showcmd
6
7 " use indentation of previous line
8 set autoindent
9 " use intelligent indentation for C
10 set smartindent
11 " configure tabwidth and insert spaces instead of tabs
12 set tabstop=4          " tab width is 4 spaces
13 set expandtab          " expand tabs to spaces
14 set showmatch
15 " intelligent comments
16 set comments=s1:/*,mb:\ *,eLx:\ */
17 set backspace=indent,eol,start
18 set softtabstop=4
19 set shiftwidth=4
20
21 map <F9> <ESC>:w<CR>:!g++ % -o %< -O2 -std=c++0x<CR>
22 map <S-F9> <ESC>:w<CR>:!g++ % -o %< -O2 -D_DEBUG_ -std=c++0x<CR>
23 map <F5> <ESC>:!. /%<<CR>
24 map <F6> <ESC>:w<CR>ggvG"+y
25 map <S-F5> <ESC>:!. /%< < %<.in<CR>
26 imap <Home> <ESC>^i
27 com INPUT sp %<.in

```

2 math

2.1 ext gcd

```

1 // find one solution (x,y) of ax+by=gcd(a,b)
2 void ext_gcd(int a,int b,int &g,int &x,int &y)
3 {
4     if(!b){ g=a; x=1; y=0; }
5     else{ ext_gcd(b, a%b, g, y, x); y -= x*(a/b); }
6 }

```

3 flow

3.1 dinic

```

1 #include <bits/stdc++.h>
2 #define PB push_back
3 #define MP make_pair
4 #define F first
5 #define S second
6 #define SZ(x) ((int)(x).size())
7 using namespace std;
8 typedef long long ll;
9 typedef pair<int,int> PII;
10 typedef vector<int> VI;
11
12 /*****/
13 // dinic
14 const int MAXV=300;
15 const int MAXE=10000;
16 const int INF=(int)1e9+10;
17
18 struct E{
19     int to,co;//capacity
20     E(int t=0,int c=0):to(t),co(c){}
21 }eg[2*MAXE];
22
23 // source:0 sink:n-1
24 struct Flow{
25     VI e[MAXV];
26     int ei,v;
27     void init(int n) {
28         v=n;
29         ei=0;
30         for(int i=0;i<n;i++)
31             e[i]=VI();
32     }
33     void add(int a,int b,int c) { //a to b ,maxflow=c
34         eg[ei]=E(b,c);
35         e[a].PB(eg[ei]);
36         ei++;
37         eg[ei]=E(a,0);
38         e[b].PB(eg[ei]);
39         ei++;
40     }
41
42     int d[MAXV],qu[MAXV],ql,qr;
43     bool BFS() {
44         memset(d,-1,v*sizeof(int));
45         ql=qr=0;
46         qu[qr++]=0;
47         d[0]=0;
48         while(ql<qr && d[v-1]==-1) {
49             int n=qu[ql++];
50             VI &v=e[n];
51             for(int i=v.size()-1;i>=0;i--) {

```

```

52     int u=v[i];
53     if(d[eg[u].to]==-1 && eg[u].co>0) {
54         d[eg[u].to]=d[n]+1;
55         qu[qr++]=eg[u].to;
56     }
57 }
58 }
59 return d[v-1]!=-1;
60 }
61 int ptr[MAXV];
62 int go(int n,int p) {
63     if(n==v-1)
64         return p;
65     VI &u=e[n];
66     int temp;
67     for(int i=ptr[n];i<SZ(u);i++)
68     {
69         if(d[n]+1!=d[eg[u[i]].to] || eg[u[i]].co==0)
70             continue;
71         if((temp=go(eg[u[i]].to,min(p,eg[u[i]].co)))==0)
72             continue;
73         eg[u[i]].co-=temp;
74         eg[u[i]^1].co+=temp;
75         ptr[n]=i;
76         return temp;
77     }
78     ptr[n]=SZ(u);
79     return 0;
80 }
81 int max_flow() {
82     int ans=0,temp;
83     while(BFS()) {
84         for(int i=0;i<v;i++)
85             ptr[i]=0;
86         while((temp=go(0,INF))>0)
87             ans+=temp;
88     }
89     return ans;
90 }
91 }flow;
92
93 int main() {
94
95     return 0;
96 }

```

4 string

4.1 KMP

```

1  /**
2  Test OJ 265
3  trivial string matching
4
5  input:
6  abc
7  abccbabbabc
8
9  output:
10 0 8
11
12 ***/

```

```

13 #include <bits/stdc++.h>
14 #define PB push_back
15 #define F first
16 #define S second
17 #define SZ(x) ((int)(x).size())
18 #define MP make_pair
19 using namespace std;
20 typedef long long ll;
21 typedef pair<int,int> PII;
22 typedef vector<int> VI;
23
24 char S[500010],T[500010];
25 int K[500010];
26
27 int main()
28 {
29     gets(S);
30     gets(T);
31     K[0]=-1;
32     int a=-1;
33     for(int i=1;S[i];i++)
34     {
35         while(a!=-1 && S[a+1]!=S[i])
36             a=K[a];
37         if(S[a+1]==S[i])
38             a++;
39         K[i]=a;
40     }
41     VI ans;
42     a=-1;
43     for(int i=0;T[i];i++)
44     {
45         while(a!=-1 && S[a+1]!=T[i])
46             a=K[a];
47         if(S[a+1]==T[i])
48             a++;
49         if(!S[a+1])
50         {
51             ans.PB(i-a);
52             a=K[a];
53         }
54     }
55     bool first=1;
56     for(int u:ans)
57     {
58         if(first)
59             printf("%d",u),first=0;
60         else
61             printf(" %d",u);
62     }
63     puts("");
64     return 0;
65 }

```

4.2 Z-value

```

1 /**
2  Test OJ 265
3  trivial string matching
4
5  input:
6  abc
7  abccbabbabc

```

```

8
9 output:
10 0 8
11
12 ***/
13 #include <bits/stdc++.h>
14 #define pb push_back
15 #define F first
16 #define S second
17 #define SZ(x) ((int)(x).size())
18 #define MP make_pair
19 using namespace std;
20 typedef long long ll;
21 typedef pair<int,int> PII;
22 typedef vector<int> VI;
23
24 char S[1000010];
25 int Z[1000010];
26
27 int main()
28 {
29     int len=0,lenS;
30     gets(S);
31     for(;S[len];len++);
32     lenS=len;
33     gets(S+len+1);
34     for(len++;S[len];len++);
35     S[len]='*';
36     int bst=0;
37     Z[0]=0;
38     for(int i=1;i<len;i++)
39     {
40         if(Z[bst]+bst<i) Z[i]=0;
41         else Z[i]=min(Z[bst]+bst-i,Z[i-bst]);
42         while(S[Z[i]]==S[i+Z[i]]) Z[i]++;
43         if(Z[i]+i>Z[bst]+bst) bst=i;
44     }
45     bool first=1;
46     for(int i=lenS+1;i<len;i++)
47         if(Z[i]>=lenS)
48         {
49             if(first)
50                 printf("%d",i-lenS-1),first=0;
51             else
52                 printf(" %d",i-lenS-1);
53         }
54     puts("");
55     return 0;
56 }

```

4.3 Suffix Array($N \log N$)

```

1 // NTUJ448
2 #include <bits/stdc++.h>
3 #define pb push_back
4 #define F first
5 #define S second
6 #define SZ(x) ((int)(x).size())
7 #define MP make_pair
8 using namespace std;
9 typedef long long ll;
10 typedef pair<int,int> PII;
11 typedef vector<int> VI;

```

```

12
13 const int SASIZE=2500000;
14 char in[500];
15 int S[SASIZE],from[SASIZE];
16 int R[SASIZE],SA[SASIZE],H[SASIZE];
17 int tR[SASIZE],tSA[SASIZE];
18 int cnt[SASIZE];
19 int num[4010];
20
21 int main()
22 {
23     int N;
24     while(scanf("%d",&N)==1 && N)
25     {
26         int len=0,maxR=0;
27         for(int i=0;i<N;i++)
28         {
29             scanf("%s",in);
30             for(int j=0;in[j];j++)
31             {
32                 from[len]=i;
33                 S[len++]=in[j]-'a';
34             }
35             from[len]=N;
36             S[len++]=i+50;
37         }
38         memset(R,-1,sizeof(R));
39         memset(cnt,0,sizeof(cnt));
40         for(int i=0;i<len;i++)
41         {
42             R[i]=S[i];
43             maxR=max(maxR,R[i]);
44         }
45         for(int i=0;i<len;i++)
46             cnt[R[i]+1]++;
47         for(int i=1;i<=maxR;i++)
48             cnt[i]+=cnt[i-1];
49         for(int i=0;i<len;i++)
50             SA[cnt[R[i]]++]=i;
51         /* for(int i=0;i<len;i++)
52            printf("R[%d]=%d, SA[%d]=%d\n",i,R[i],i,SA[i]); */
53         for(int i=1;i<len;i*=2)
54         {
55             memset(cnt,0,sizeof(int)*(maxR+10));
56             memcpy(tSA,SA,sizeof(int)*(len+10));
57             memcpy(tR,R,sizeof(int)*(len+i+10));
58             for(int j=0;j<len;j++)
59                 cnt[R[j]+1]++;
60             for(int j=1;j<=maxR;j++)
61                 cnt[j]+=cnt[j-1];
62             for(int j=len-i;j<len;j++)
63                 SA[cnt[R[j]]++]=j;
64             for(int j=0;j<len;j++)
65             {
66                 int k=tSA[j]-i;
67                 if(k<0)
68                     continue;
69                 SA[cnt[R[k]]++]=k;
70             }
71             int num=0;
72             maxR=0;
73             R[SA[0]]=num;
74             for(int j=1;j<len;j++)
75             {
76                 if(tR[SA[j-1]]<tR[SA[j]] || tR[SA[j-1]+i]<tR[SA[j]+i])

```

```

77     num++;
78     R[SA[j]]=num;
79     maxR=max(maxR,R[SA[j]]);
80 }
81 /* puts("-----");
82 for(int i=0;i<len;i++)
83     printf("R[%d]=%d, SA[%d]=%d\n",i,R[i],i,SA[i]);*/
84 }
85 memset(H,0,sizeof(H));
86 for(int i=0;i<len;i++)
87 {
88     if(R[i]==0)
89         continue;
90     int &t=H[R[i]];
91     if(i>0)
92         t=max(0,H[R[i-1]]-1);
93     while(S[i+t]==S[SA[R[i]-1]+t]) t++;
94 }
95 /*for(int i=0;i<len;i++)
96     printf("R[%d]=%d, SA[%d]=%d\n",i,R[i],i,SA[i]);
97 for(int i=0;i<len;i++)
98     printf("%3d %3d %s\n",H[i],SA[i],S+SA[i]);*/
99 /*for(int i=0;i<len;i++)
100 {
101     printf("%3d %3d %d|",H[i],SA[i],from[i]);
102     for(int j=SA[i];j<len;j++)
103         printf("%2d ",S[j]);
104     puts("");
105 }*/
106 memset(num,0,sizeof(num));
107 int anslen=0,ansfrom=-1;
108 int get=0;
109 deque<PII> deq;
110 /* for(int i=0;i<len;i++)
111     printf("%d:%d\n",i,from[i]);*/
112 for(int l=0,r=0;r<len;r++)
113 {
114     if(from[SA[r]]<N && num[from[SA[r]]]==0)
115         get++;
116     num[from[SA[r]]]++;
117     while(deq.size()>0 && deq.back().F>=H[r]) deq.pop_back();
118     deq.pb(MP(H[r],r));
119     while(num[from[SA[l]]]>1)
120     {
121         num[from[SA[l]]]--;
122         l++;
123     }
124     while(deq.size()>0 && deq.front().S<=1) deq.pop_front();
125     if(get==N && deq.front().F>anslen)
126         anslen=deq.front().F, ansfrom=SA[l];
127 }
128 //printf("(%d)\n",anslen);
129 if(anslen==0)
130     puts("IDENTITY LOST");
131 else
132 {
133     for(int i=ansfrom;i<ansfrom+anslen;i++)
134         putchar(S[i]+'a');
135     puts("");
136 }
137 }
138 return 0;
139 }

```


5 graph

6 data structure

7 geometry