

ACM Notebook team HCMUS-KMN

KMN

October 4, 2019

1 Some define

```
1 #include <bits/stdc++.h>
2
3 #define maxN
4 #define matrix_size 2
5 #define base 1000000007LL
6 #define eps 1e-8
7
8 #define cross(A,B) (A.x*B.y-A.y*B.x)
9 #define dot(A,B) (A.x*B.x+A.y*B.y)
10 #define ccw(A,B,C) (-(A.x*(C.y-B.y) + B.x*(A.y-C.y) + C.x*(B.y-A.y))) // positive
    when ccw
11 #define CROSS(a,b,c,d) (a*d - b*c)
```

2 Geometry

```
1 struct line
2 {
3     double a,b,c;
4     line() {}
5     line(double A,double B,double C):a(A),b(B),c(C){}
6     line(Point A,Point B)
7     {
8         a=A.y-B.y; b=B.x-A.x; c=-a*A.x-b*A.y;
9     }
10 };
11
12 Point intersect(line AB,line CD)
13 {
14     AB.c=-AB.c; CD.c=-CD.c;
15     double D=CROSS(AB.a,AB.b,CD.a,CD.b);
16     double Dx=CROSS(AB.c,AB.b,CD.c,CD.b);
17     double Dy=CROSS(AB.a,AB.c,CD.a,CD.c);
18     if (D==0.0) return Point(1e9,1e9);
19     else return Point(Dx/D,Dy/D);
20 }
```

3 Dinic

```
1 bool BFS()
2 {
3     queue<int> q;
4     for (int i=1; i<=n; i++) d[i]=0,Free[i]=true;
5     q.push(s);
6     d[s]=1;
7     while (!q.empty())
8     {
9         int u=q.front(); q.pop();
10        for (int i=0; i<DSK[u].size(); i++)
11        {
12            int v=DSK[u][i].fi;
13            if (d[v]==0 && DSK[u][i].se>f[u][v])
14            {
15                d[v]=d[u]+1;
16                q.push(v);
17            }
18        }
19    }
```

```

19     }
20     return d[t]!=0;
21 }
22
23 int DFS(int x,int delta)
24 {
25     if (x==t) return delta;
26     Free[x]=false;
27     for (int i=0; i<DSK[x].size(); i++)
28     {
29         int y=DSK[x][i].fi;
30         if (d[y]==d[x]+1 && f[x][y]<DSK[x][i].se && Free[y])
31         {
32             int tmp=DFS(y,min(delta,DSK[x][i].se-f[x][y]));
33             if (tmp>0)
34             {
35                 f[x][y]+=tmp; f[y][x]-=tmp; return tmp;
36             }
37         }
38     }
39     return 0;
40 }

```

4 Mincost

```

1 int calc(int x,int y){ return (x>=0) ? y : 0-y; }
2
3 bool findpath()
4 {
5     for (int i=1; i<=n; i++){ trace[i]=0; d[i]=inf; } q.push(n); d[n]=0;
6     while (!q.empty())
7     {
8         int u=q.front(); q.pop(); inq[u]=false;
9         for (int i=0; i<DSK[u].size(); i++)
10        {
11            int v=DSK[u][i];
12            if (c[u][v]>f[u][v] && d[v]>d[u]+calc(f[u][v],cost[u][v]))
13            {
14                trace[v]=u;
15                d[v]=d[u]+calc(f[u][v],cost[u][v]);
16                if (!inq[v])
17                {
18                    inq[v]=true;
19                    q.push(v);
20                }
21            }
22        }
23    }
24    return d[t]!=inf;
25 }
26
27 void incflow()
28 {
29     int v=t,delta=inf;
30     while (v!=n)
31     {
32         int u=trace[v];
33         if (f[u][v]>=0) delta=min(delta,c[u][v]-f[u][v]);
34         else delta=min(delta,0-f[u][v]);

```

```

35     v=u;
36 }
37 v=t;
38 while (v!=n)
39 {
40     int u=trace[v];
41     f[u][v]+=delta; f[v][u]-=delta;
42     v=u;
43 }
44 }

```

5 HLD

```

1 void DFS(int x,int pa)
2 {
3     DD[x]=DD[pa]+1; child[x]=1; int Max=0;
4     for (int i=0; i<DSK[x].size(); i++)
5     {
6         int y=DSK[x][i].fi;
7         if (y==pa) continue;
8         p[y]=x;
9         d[y]=d[x]+DSK[x][i].se;
10        DFS(y,x);
11        child[x]+=child[y];
12        if (child[y]>Max)
13        {
14            Max=child[y];
15            tree[x]=tree[y];
16        }
17    }
18    if (child[x]==1) tree[x]=++nTree;
19 }
20
21 void init()
22 {
23     nTree=0;
24     DFS(1,1);
25     DD[0]=long(1e9);
26     for (int i=1; i<=n; i++) if (DD[i]<DD[root[tree[i]]]) root[tree[i]]=i;
27 }
28
29 int LCA(int u,int v)
30 {
31     while (tree[u]!=tree[v])
32     {
33         if (DD[root[tree[u]]]<DD[root[tree[v]]]) v=p[root[tree[v]]];
34         else u=p[root[tree[u]]];
35     }
36     if (DD[u]<DD[v]) return u; else return v;
37 }

```

6 Cầu khớp

Nút u là khớp: if (low[v] >= num[u]) arti[u] = arti[u] || p[u] != -1 || child[u] >= 2;
 Cạnh u, v là cầu khi low[v] >= num[v]

7 Monotone chain

```
1 void convex_hull (vector<pt> & a) {
2     if (a.size() == 1) { // ich có 1 điểm
3         return;
4     }
5
6     // Sort with respect to x and then y
7     sort(a.begin(), a.end(), &cmp);
8
9     pt p1 = a[0], p2 = a.back();
10
11     vector<pt> up, down;
12     up.push_back (p1);
13     down.push_back (p1);
14
15     for (size_t i=1; i<a.size(); ++i) {
16         // Add to the upper chain
17
18         if (i==a.size()-1 || cw (p1, a[i], p2)) {
19             while (up.size()>=2 && !cw (up[up.size()-2], up[up.size()-1], a[i]))
20                 up.pop_back();
21             up.push_back (a[i]);
22         }
23
24         // Add to the lower chain
25         if (i==a.size()-1 || ccw (p1, a[i], p2)) {
26             while (down.size()>=2 && !ccw (down[down.size()-2], down[down.size()-1], a[i]))
27                 down.pop_back();
28             down.push_back (a[i]);
29         }
30     }
31
32     // Merge 2 chains
33     a.clear();
34     for (size_t i=0; i<up.size(); ++i)
35         a.push_back (up[i]);
36     for (size_t i=down.size()-2; i>0; --i)
37         a.push_back (down[i]);
38 }
```

8 MST

Prim: remember to have visited array

9 Bignum mul

```
1 string mul(string a,string b)
2 {
3     int m=a.length(),n=b.length(),sum=0;
4     string c="";
5     for (int i=m+n-1; i>=0; i--)
6     {
7         for (int j=0; j<m; j++) if (i-j>0 && i-j<=n) sum+=(a[j]-'0')*(b[i-j-1]-'0');
8         c=(char)(sum%10+'0')+c;
```

```

9      sum/=10;
10    }
11    while (c.length()>1 && c[0]=='0') c.erase(0,1);
12    return c;
13 }

```

10 Prime under 100

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

11 Pascal triangle $C(n,k)$ =number from line 0, column 0

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1

```

12 Fibo

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

13 Tips

1. Giả sử nó là số nguyên tố đi. Giả sử nó liên quan tới số nguyên tố đi.
2. Giả sử nó là số có dạng 2^n đi.
3. Giả sử chọn tối đa là 2, 3 số gì là có đáp án đi.
4. Có liên quan gì tới Fibonacci hay tam giác pascal?
5. Dãy này đơn điệu không em ei? Hay tổng của 2,3 số fibonacci?
6. $q \leq 2$
7. Sort lại đi, biết đâu thấy điều hay hơn?
8. Chia nhỏ ra xem.

9. Bỏ hết những thằng ko cần thiết ra
10. Áp đại data struct nào đấy vô
11. khóc
12. Cầu nguyện
13. Random shuffe để ac
14. Xoay mảng 45 độ