

1 Computational Geometry

1.1 SmallestCircle

```

1 using PT=point<T>; using CPT=const PT;
2 PT circumcenter(CPT &a,CPT &b,CPT &c){
3     PT u=b-a, v=c-a;
4     T c1=u.abs2()/2,c2=v.abs2()/2;
5     T d=u.cross(v);
6     return PT(a.x+(v.y*c1-u.y*c2)/d,a.y+(u.
7         x*c2-v.y*c1)/d);
8 }
9 void solve(PT p[],int n,PT &c,T &r2){
10     random_shuffle(p,p+n);
11     c=p[0]; r2=0; // c,r2 = 圓心,半徑平方
12     for(int i=1;i<n;i++){
13         if((p[i]-c).abs2()>r2)
14             c=p[i]; r2=0;
15         for(int j=0;j<i;j++){
16             if((p[j]-c).abs2()>r2)
17                 c.x=(p[i].x+p[j].x)/2;
18                 c.y=(p[i].y+p[j].y)/2;
19                 r2=(p[i]-c).abs2();
20         }
21     }
22 }
23 }

```

1.2 最近點對

```

1 template<typename _IT=point<T>*>
2 T closest_pair(_IT L, _IT R){
3     if(R-L <= 1) return INF;
4     _IT mid = L+(R-L)/2;
5     T x = mid->x;
6     T d = min(closest_pair(L,mid),
7         closest_pair(mid,R));
8     inplace_merge(L, mid, R, ycmp);
9     static vector<point> b; b.clear();
10     for(auto u=L;u<R;++u){
11         if((u->x-x)*(u->x-x)>=d) continue;
12         for(auto v=b.rbegin();v!=b.rend();++v)
13             if(dx=u->x-v->x, dy=u->y-v->y;
14                 if(dy*dy>=d) break;
15                 d=min(d,dx*dx+dy*dy);
16     }
17     b.push_back(*u);
18     return d;
19 }
20 T closest_pair(vector<point<T>> &v){
21     sort(v.begin(),v.end(),xcmp);
22     return closest_pair(v.begin(),v.end());
23 }

```

2 Data Structure

2.1 01 背包

```

1 LL dp[101][100001] = {0}; //前i個物品所湊
2 //出重量j的最大價值
3 int main(){
4     good;
5     LL j,i,n,w,svalue = 0,sweight = 0;
6     cin >> n >> w;
7     pair<LL,LL> item[n+1]; //weight,value;
8     for(i=1;i<=n;i++){
9         cin >> item[i].first;
10        for(i=1;i<=n;i++){
11            cin >> item[i].second;
12        }
13        for(i=0;i<=n;i++){
14            dp[i][0] = dp[0][i] = 0;
15        }
16        for(i=1;i<=n;i++){
17            for(j=1;j<=w;j++){
18                if(item[i].first > j)
19                    dp[i][j] = dp[i-1][j];
20                else
21                    dp[i][j] = max(dp[i-1][j],
22                        item[i].second +
23                        dp[i-1][j-item[i].
24                            first]);
25            }
26        }
27    }

```

```

19 }
20 }
21 cout << dp[n][w];
22 return 0;
23 }

```

2.2 array simulate linked list

```

1 #include<bits/stdc++.h>
2 #define pll pair<LL,LL>
3 #define RSIZE 100002
4 #define oo 1000000001
5 #define good ios_base::sync_with_stdio(0)
6 ;cin.tie(0)
7 typedef long long LL;
8 using namespace std;
9 struct {
10     LL center,height;
11     LL pre,next;
12     bool alive;
13 }tree[RSIZE];
14 queue<LL> Q; //check removed tree
15 void removable(LL index){
16     if(!tree[index].alive) return;
17     LL s = tree[index].pre,t = tree[index].
18         next;
19     if(tree[index].center-tree[s].
20         center >= tree[s].center ||
21         tree[index].center+tree[t].
22         center <= tree[t].center){
23         tree[index].alive = false;
24         Q.push(index);
25         tree[s].next = t;
26         tree[t].pre = s;
27     }
28 }
29 int main(){
30     good;
31     //freopen("P_2_1_5.in", "r", stdin)
32     //input redirection
33     LL bound,n,total = 0,height = 0;
34     cin >> n >> bound;
35     for(LL i=1;i<=n;i++){
36         cin >> tree[i].center;
37         for(LL i=1;i<=n;i++){
38             cin >> tree[i].height;
39             for(LL i=1;i<=n;i++){
40                 tree[i].pre = i-1;
41                 tree[i].next = i+1;
42                 tree[i].alive = true;
43             }
44         }
45         tree[0].center = 0,tree[0].height =
46         oo;
47         tree[n+1].center = bound,tree[n+1].
48         height = oo;
49         for(LL i=1;i<=n;i++){
50             removable(i);
51             while (!Q.empty()){
52                 LL v = Q.front();
53                 Q.pop();
54                 total++;
55                 high = max(high,tree[v].
56                     height);
57                 removable(tree[v].pre);
58                 removable(tree[v].next);
59             }
60             cout << total << endl << high;
61             return 0;
62         }
63     }

```

2.3 binary search

```

1 LL BS(LL left,LL right){
2     if(left+1 >= right) //break condition
3         return -1;
4     LL mid = (left+right)/2;
5     if(arr[mid] == target)
6         return mid;
7     else if(arr[mid] < target){
8         left = mid+1;
9         BS(left,right);
10    }
11    else if(arr[mid] > target){
12        right = mid;
13        BS(left,right);
14    }
15 }

```

2.4 discretization

```

1 map<LL,LL> S;
2 for (LL i=0;i<n;i++){
3     S[a[i]] = 0; // insert a[i] and
4     set rank=0
5     LL r=0;
6     for (auto it=S.begin(); it!=S.end();
7         ++it) //traversal and set rank
8         it->second = r++;
9     // replace number with rank
10    for (LL i=0;i<n;i++){
11        a[i] = S.lower_bound(a[i]) ->
12            second;
13    }
14    // find() return the iterator,
15    // then take the rank
16    // or S.find(a[i]) -> second;
17 }

```

2.5 half enumeration

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 ;cin.tie(0)
4 typedef long long LL;
5 using namespace std;
6 LL sa[1<<18],sb[1<<18],no[1<<18]; //subset
7 //product of a and b
8 LL subset(LL num[],LL length,LL product
9     [],LL p){ //pass by pointer
10    LL k = 0,i,j; //count
11    for(i=0;i<length;i++){
12        for(j=0;j<k;j++){
13            product[k+j] = (product[j]*
14                num[i]) % p; //old
15            //product times num[i]
16        }
17        product[k] = num[i]; //for num[i]
18        //itself
19        k += k+1;
20    }
21    return k; //return the size of subset
22 }
23 LL exp_modp(LL x,LL y,LL p){
24    if(y == 0) return 1;
25    if(y % 2) return (exp_modp(x,y-1,p)*x
26        ) % p;
27    else{
28        LL temp = exp_modp(x,y/2,p);
29        return (temp*temp) % p;
30    }
31 }
32 int main(){
33     good;
34     //freopen("file name", "r", stdin);
35     //input redirection
36     LL i,n,p;
37     LL a[30],b[30];
38     cin >> n >> p;
39     int len_a = n/2,len_b = n - len_a;
40     for(i=0;i<len_a;i++){
41         cin >> a[i];
42     }
43     for(i=0;i<len_b;i++){
44         cin >> b[i];
45     }
46     LL len_sa = subset(a,len_a,sa,p);
47     LL len_sb = subset(b,len_b,sb,p);
48     sort(sa,sa+len_sa);
49     sort(sb,sb+len_sb);
50     LL len_sb2 = 1; //len_sb2 followed by
51     //i below
52     no[0] = 1; //assume not empty(check
53     //later)
54     for(i=1;i<len_sb;i++){
55         if(sb[i] != sb[i-1]){ //new
56             //element
57             sb[len_sb2] = sb[i];
58             no[len_sb2] = 1;
59             len_sb2++;
60         }
61         else //old element
62             no[len_sb2-1]++;
63     }
64     LL ans = (sb[0] == 1) ? no[0] % p : 0;
65     for(i=0;i<len_sa;i++){
66         if(sa[i] == 1) ans = (ans+1) % p;
67         LL y = exp_modp(sa[i],p-2,p); //
68         //module inverse
69         int it = lower_bound(sb,sb+
70             len_sb2,y) - sb;
71     }

```

```

60         if(it < len_sb2 && sb[it] == y){
61             ans = (ans + no[it]) % p;
62         }
63     }
64     cout << ans << '\n';
65     return 0;
66 }

```

2.6 LCS

```

1 int dp[1002][1002],i,j; //text1 前i個 &
2   text2 前j個
3 for(i = 0; i < 1002; i++)
4   dp[i][0] = 0,dp[0][i] = 0;
5 for(i = 1; i <= text1.size(); i++){
6   //1 base <=
7   for(j = 1; j <= text2.size(); j
8     ++){
9     if(text1[i-1] == text2[j-1])
10      dp[i][j] = dp[i-1][j-1]+1;
11     else
12      dp[i][j] = max(dp[i-1][j],dp[i][j-1]);
13 }
14 cout << dp[text1.size()][text2.size()]
15 ;

```

2.7 LIS

```

1 int main(){
2   good;
3   //freopen("file name", "r", stdin);
4   //input redirection
5   LL n,i,length = 0,num;
6   cin >> n;
7   LL last[RSIZE]; //長度為it的最小可能結
8   尾
9   for(i = 0; i < n; i++){
10    cin >> num;
11    LL it = lower_bound(last,last+
12      length,num)-last;
13    last[it] = num;
14    if(it == length) length++;
15 }
16 cout << length;
17 return 0;

```

2.8 skew heap

```

1 node *merge(node *a,node *b){
2   if(!a||!b) return a:b;
3   if(b->data<a->data) swap(a,b);
4   swap(a->l,a->r);
5   a->l=merge(b,a->l);
6   return a;
7 }

```

2.9 sliding window

```

1 //same size
2 for(i = 0; i < m; i++){//making first
3   window
4   LL color = discret[a[right]];
5   cnt[color]++;
6   if(cnt[color] == 1) n_color++;
7   right++;
8 }
9 while(right < n){
10  if(n_color == m)
11    ans++;
12  LL l_remove = discret[a[left]];
13  cnt[l_remove]--; //remove left one
14  left++;
15  if(cnt[l_remove] == 0) n_color--;
16  LL add = discret[a[right]];
17  cnt[add]++; right++; //add next one
18  if(cnt[add] == 1) n_color++;
19 }

```

2.10 undo disjoint set

```

1 struct DisjointSet {
2   // save() is like recursive
3   // undo() is like return
4   int n, fa[MXN], sz[MXN];
5   vector<pair<int*,int>> h;
6   vector<int> sp;
7   void init(int tn) {
8     n=tn;
9     for (int i=0; i<n; i++) sz[fa[i]=i]
10      =1;
11     sp.clear(); h.clear();
12 }
13 void assign(int *k, int v) {
14   h.PB({k, *k});
15   *k=v;
16 }
17 void save() { sp.PB(SZ(h)); }
18 void undo() {
19   assert(!sp.empty());
20   int last=sp.back(); sp.pop_back();
21   while (SZ(h)!=last) {
22     auto x=h.back(); h.pop_back();
23     *x.F=x.S;
24 }
25 int f(int x) {
26   while (fa[x]!=x) x=fa[x];
27   return x;
28 }
29 void uni(int x, int y) {
30   x=f(x); y=f(y);
31   if (x==y) return ;
32   if (sz[x]<sz[y]) swap(x, y);
33   assign(&sz[x], sz[x]+sz[y]);
34   assign(&fa[y], x);
35 }
36 }djs;

```

3 Graph

3.1 bellman ford

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3   ;cin.tie(0)
4 #define RSIZE 101
5 #define pll pair<LL,LL>
6 #define lc 2*index
7 #define rc 2*index+1
8 #define maxdis 1000000009
9 typedef long long LL;
10 using namespace std;
11 struct Node{
12   LL in,out,weight;
13 };
14 LL dis[RSIZE];
15 void bellman_ford(vector<Node> &edges,LL
16   v,LL e,LL start){//v for vertex, e
17   for edge
18   //vector<LL> dis(v,LLONG_MAX);
19   dis[start] = 0;
20   //relaxation
21   for(LL i = 0; i < v-1; i++){
22     for(LL j = 0; j < e; j++){
23       LL x = edges[j].in,y = edges[
24         j].out;
25       LL w = edges[j].weight;
26       if(dis[x] != maxdis && dis[x]
27         +w < dis[y])
28         dis[y] = dis[x] + w;
29     }
30   }
31   //check -weight round
32   cout << "from " << start << "'s
33     shortest path to each vertexes
34     is:" << endl;
35   for(LL i = 0; i < v; i++){
36     cout << i << "\t\t" << dis[i] <<
37     endl;
38 }
39 bool checkinf(vector<Node> &edges,LL e){
40   for(LL i = 0; i < e; i++){
41     LL x =edges[i].in,y = edges[i].
42     out;
43     LL w = edges[i].weight;
44     if(dis[x] != maxdis && dis[x] + w
45       < dis[y])
46       return true;
47 }

```

```

38 }
39 return false;
40 }
41 int main(){
42   good;
43   LL vertex,edge;
44   cin >> vertex >> edge;
45   memset(dis,maxdis,sizeof(dis));
46   vector<Node> graph(edge);
47   for(LL i = 0; i < edge; i++){
48     cin >> graph[i].in >> graph[i].
49     out >> graph[i].weight;
50 }
51 LL source;
52 cin >> source;
53 bellman_ford(graph,vertex,edge,source
54 );
55 if(checkinf(graph,edge))
56   cout << "found negative round" <<
57   endl;
58 else
59   cout << "not found negative round
60   " << endl;
61 return 0;

```

3.2 BFS

```

1 LL val;//unnecessary
2 bool visited[5000] = {false};
3 vector<LL> graph[5000];
4 void BFS(LL start) {
5   queue<LL> q;
6   q.push(start);
7   visited[start] = true;
8   while (!q.empty()){
9     LL curr = q.front();
10    q.pop();
11    for(auto it: graph[curr]){
12      if(!visited[it]){
13        q.push(it);
14        visited[it] = true;
15      }
16    }
17 }
18 }

```

3.3 DFS

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3   ;cin.tie(0);cout.tie(0)
4 typedef long long LL;
5 using namespace std;
6 int fa[100000],d[100000] = {0}; //
7 unnecessary
8 bool visit[100000] = {false};
9 vector<LL> v[100000];
10 void dfs(LL now,LL depth){
11   for(auto x:v[now]){
12     if(!visit[x]){
13       cout << x << ' ';
14       visit[x] = true;
15       d[x] = depth;
16       fa[x] = now;
17       dfs(x,depth+1);
18     }
19 }
20 int main(){
21   good;
22   LL i,n,a,b;
23   cin >> n;
24   for(i = 0; i < n; i++){
25     cin >> a >> b;
26     v[a].push_back(b);
27     v[b].push_back(a);
28 }
29 dfs(0,1);
30 return 0;

```

3.4 dijkstra

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 #define N 10002
4 #define oo 1000000001//1e9+1
5 typedef long long LL;
6 using namespace std;
7
8 vector<pair<LL,LL>> adjacent[N];//out
9 neighbor,weight of edge
10 LL dis[N],parent[N];
11 bool visit[N] = {false};
12
13 int main(){
14     LL i,n,m;
15     cin >> n >> m;
16     for(i = 0; i < m; i++){
17         LL x,y,w;
18         cin >> x >> y >> w;
19         adjacent[x].push_back({y,w});
20         adjacent[y].push_back({x,w});
21     }
22     //initial
23     LL source = 0;
24     memset(dis,oo,sizeof(dis));
25     memset(parent,-1,sizeof(parent));
26     priority_queue<pair<LL,LL>> PQ;//-dis
27     [],vertex · 技巧性讓最小值pop
28     PQ.push({dis[source] = 0,source});
29     //dijkstra
30     while (!PQ.empty()){
31         auto p = PQ.top();
32         PQ.pop();
33         LL v = p.second;//vertex
34         if(visit[v]) continue;
35         visit[v] = true;
36         for(auto it : adjacent[v]){
37             LL e = it.first,w = it.second;
38             if(w + dis[v] < dis[e]){
39                 dis[e] = w + dis[v];
40                 parent[e] = v;
41                 PQ.push({-dis[e],e});
42             }
43         }
44         LL maxd = -1,cnt = 0,far;
45         for(i = 0; i < n; i++){
46             if(dis[i] < oo){
47                 if(dis[i] > maxd)
48                     maxd = dis[i],far = i;
49             }
50             else
51                 cnt++;//for can't reach
52         }
53         cout << maxd << endl << cnt;
54         return 0;
55     }
56 }

```

3.5 topology sort

```

1 int main(){
2     good;
3     LL indeg[1002] = {0};
4     vector<LL> graph[1002];
5     LL n,m,a,b;
6     cin >> n >> m;
7     for(LL i = 0; i < m; i++){
8         cin >> a >> b;
9         graph[a].push_back(b);
10        indeg[b]++;
11    }
12    LL topo[1002],head = 0,tail = 0;//??
13    queue
14    for(LL i = 0; i < n; i++){
15        if(indeg[i] == 0)
16            topo[tail++] = i;
17    }
18    while(head < tail){
19        LL v = topo[head++];//get data
20        and pop
21        for(LL u : graph[v]){
22            if(--indeg[u] == 0)
23                topo[tail++] = u;
24        }
25    }
26    if(tail < n) cout << "not a DAG" << endl;
27    else{
28        for(LL i = 0; i < n; i++){
29            cout << topo[i] << ' ';
30        }
31    }
32 }

```

3.6 union and find

```

28     return 0;
29 }
30
31 #include<bits/stdc++.h>
32 #define good ios_base::sync_with_stdio(0)
33 #define RSIZE 101
34 typedef long long LL;
35 using namespace std;
36
37 LL parent[503*503];
38 int graph[503*503] = {0};
39 int dxy[4] = {1,-1};
40 LL now_area = 0,max_area = 0;
41
42 LL sfind(LL dots){//find Leader,Leader's
43     parent = size of set
44     if(parent[dots] < 0)
45         return dots;
46     return parent[dots] = sfind(parent[
47         dots]);
48 }
49
50 LL BFS(LL now,LL root){//find root and
51     return size
52     parent[now] = root;
53     LL cnt = 1;
54     for(int k = 0; k < 4; k++){//4
55         directions
56         int u = now+dxy[k];
57         if(graph[u] == 1 && parent[u] ==
58             -1)//unvisited
59             cnt += BFS(u,root);
60     }
61     return cnt;
62 }
63
64 void combine(LL u,LL v){//merge two sets
65     LL set1 = sfind(u),set2 = sfind(v);
66     if(set1 == set2) return;//same set
67
68     max_area = max(max_area,-parent[set1]
69         -parent[set2]);
70     now_area--;//merge -> 2 pools become
71     1
72     if(parent[set1] < parent[set2]){//1
73         is larger
74         parent[set1] += parent[set2];
75         parent[set2] = set1;
76     }
77     else{
78         parent[set2] += parent[set1];
79         parent[set1] = set2;
80     }
81     return;
82 }
83
84 int main(){
85     good;
86     //freopen("file name", "r", stdin);
87     //input redirection
88     LL i,j,m,n,k;
89     cin >> m >> n >> k;
90     memset(parent,-1,sizeof(parent));
91     for(i = 1; i <= m; i++){
92         for(j = 1; j <= n; j++){
93             cin >> graph[i*(n+2)+j];
94         }
95     }
96     n += 2;
97     dxy[2] = n,dxy[3] = -n;
98     LL mn = (m+1)*n;
99     for(LL x = n; x < mn; x++){
100        if(graph[x] == 1 && parent[x] ==
101            -1)//unvisited
102            parent[x] = -BFS(x,x);//first
103            point consider as root
104            now_area++;
105            max_area = max(max_area,-
106                parent[x]);
107        }
108    }
109    LL ans = now_area,max_ans = max_area;
110    while(k--){
111        LL x,y,temp;
112        cin >> x >> y;
113        temp = x*n+y;
114        if(graph[temp] == 1) continue;
115        graph[temp] = 1;
116        now_area++;
117        max_area = max(max_area,(LL)1);
118        for(i = 0; i < 4; i++){
119            if(graph[temp+dxy[i]] == 0)
120                continue;
121            combine(temp,temp+dxy[i]);
122        }
123    }
124 }

```

```

76     ans += now_area;
77     max_ans += max_area;
78 }
79 cout << max_ans << endl << ans;
80 return 0;
81 }

```

4 Number Theory

4.1 basic

```

1 template<typename T>
2 void gcd(const T &a,const T &b,T &d,T &x,
3     T &y){
4     if(!b) d=a,x=1,y=0;
5     else gcd(b,a%b,d,y,x), y-=x*(a/b);
6 }
7
8 long long int phi[N+1];
9 void phiTable(){
10     for(int i=1;i<=N;i++)phi[i]=i;
11     for(int i=1;i<=N;i++){
12         for(x=i*2;x<=N;x+=
13             i)phi[x]-=phi[i];
14     }
15 }
16
17 const int MAXPRIME = 1000000;
18 int iscom[MAXPRIME], prime[MAXPRIME],
19     primecnt;
20 int phi[MAXPRIME], mu[MAXPRIME];
21 void sieve(void){
22     memset(iscom,0,sizeof(iscom));
23     primecnt = 0;
24     phi[1] = mu[1] = 1;
25     for(int i=2;i<MAXPRIME;i++){
26         if(!iscom[i]){
27             prime[primecnt++] = i;
28             mu[i] = -1;
29             phi[i] = i-1;
30         }
31         for(int j=0;j<primecnt;j++){
32             int k = i * prime[j];
33             if(k>MAXPRIME) break;
34             iscom[k] = prime[j];
35             if(i%prime[j]==0){
36                 mu[k] = 0;
37                 phi[k] = phi[i] * prime[j];
38                 break;
39             }
40             else{
41                 mu[k] = -mu[i];
42                 phi[k] = phi[i] * (prime[j]-1);
43             }
44         }
45     }
46 }
47
48 bool g_test(const LL &g, const LL &p,
49     const vector<LL> &v) {
50     for(int i=0;i<v.size();i++){
51         if(modexp(g,(p-1)/v[i],p)==1)
52             return false;
53     }
54     return true;
55 }
56
57 LL primitive_root(const LL &p) {
58     if(p==2) return 1;
59     vector<LL> v;
60     Factor(p-1,v);
61     v.erase(unique(v.begin(), v.end()), v.
62         end());
63     for(LL g=2;g<p;g++){
64         if(g_test(g,p,v))
65             return g;
66     }
67     puts("primitive_root NOT FOUND");
68     return -1;
69 }
70
71 int Legendre(const LL &a, const LL &p) {
72     return modexp(a%p,(p-1)/2,p);
73 }
74
75 LL inv(const LL &a, const LL &n) {
76     LL d,x,y;
77     gcd(a,n,d,x,y);
78     return d==1 ? (x+n)%n : -1;
79 }
80
81 int inv[N];
82 LL invtable(int n,LL P){
83     inv[1]=1;
84     for(int i=2;i<=n;i++){
85         inv[i]=(P-(P/i))*inv[P%i]%P;
86     }
87 }

```

```

76 LL log_mod(const LL &a, const LL &b,
77 const LL &p) {
78 //  $a^x \equiv b \pmod p$ 
79 int m=sqrt(p+.5), e=1;
80 LL v=inv(modexp(a,m,p), p);
81 map<LL,int> x;
82 x[1]=0;
83 for(int i=1;i<m;++i) {
84 e = LLMul(e,a,p);
85 if(!x.count(e)) x[e] = i;
86 }
87 for(int i=0;i<m;++i) {
88 if(x.count(b)) return i*m + x[b];
89 b = LLMul(b,v,p);
90 }
91 return -1;
92 }
93
94 LL Tonelli_Shanks(const LL &n, const LL &
95 p) {
96 //  $x^2 \equiv n \pmod p$ 
97 if(n==0) return 0;
98 if(Legendre(n,p)!=1) while(1) { puts("
99 SQRRT ROOT does not exist"); }
100 int S = 0;
101 LL Q = p-1;
102 while( !(Q&1) ) { Q>>=1; ++S; }
103 if(S==1) return modexp(n%p,(p+1)/4,p);
104 LL z = 2;
105 for(; Legendre(z,p)!=-1; ++z)
106 LL c = modexp(z,Q,p);
107 LL R = modexp(n%p,(Q+1)/2,p), t =
108 modexp(n%p,Q,p);
109 int M = S;
110 while(1) {
111 if(t==1) return R;
112 LL b = modexp(c,1L<<(M-i-1),p);
113 R = LLMul(R,b,p);
114 t = LLMul(LLmul(b,b,p), t, p);
115 c = LLMul(b,b,p);
116 M = i;
117 }
118 return -1;
119 }
120
121 template<typename T>
122 T Euler(T n){
123 T ans=n;
124 for(T i=2;i*i<=n;++i){
125 if(n%i==0){
126 ans=ans/i*(i-1);
127 while(n%i==0)n/=i;
128 }
129 if(n>1)ans=ans/n*(n-1);
130 return ans;
131 }
132 //Chinese_remainder_theorem
133 template<typename T>
134 T pow_mod(T n,T k,T m){
135 T ans=1;
136 for(n=(n>m?n%m:n);k;k>>=1){
137 if(k&1)ans=ans*n%m;
138 n=n*n%m;
139 }
140 return ans;
141 }
142 template<typename T>
143 T crt(vector<T> &m,vector<T> &a){
144 T M=1,tM,ans=0;
145 for(int i=0;i<(int)m.size();++i)M*=m[i];
146 for(int i=0;i<(int)a.size();++i){
147 tM=M/m[i];
148 ans=(ans+(a[i]*tM%M)*pow_mod(tM,Euler(m[i]),M)%M;
149 //如果m[i]是質數·Euler(m[i])-1=m[i]
150 // -2·就不用算Euler了*/
151 }
152 return ans;
153 }
154 //java code
155 //求sqrt(N)的連分數
156 public static void Pell(int n){
157 BigInteger N,p1,p2,q1,q2,a0,a1,a2,g1,g2;
158 h1=h2=p1=BigInteger.ZERO;
159 h1=q1=p2=BigInteger.ONE;
160 a0=a1=BigInteger.valueOf((int)Math.sqrt(
161 1.0*n));
162 BigInteger ans=a0.multiply(a0);
163 if(ans.equals(BigInteger.valueOf(n))){
164 System.out.println("No solution!");
165 }

```

```

166 return ;
167 }
168 while(true){
169 g2=a1.multiply(h1).subtract(g1);
170 h2=N.subtract(g2.pow(2)).divide(h1);
171 a2=g2.add(a0).divide(h2);
172 p=a1.multiply(p2).add(p1);
173 q=a1.multiply(q2).add(q1);
174 if(p.pow(2).subtract(N.multiply(q.
175 pow(2))).compareTo(BigInteger.
176 ONE)==0)break;
177 g1=g2;h1=h2;a1=a2;
178 p1=p2;p2=p;
179 q1=q2;q2=q;
180 }
181 System.out.println(p+" "+q);
182 }

```

4.2 bit set

```

1 void sub_set(int S){
2 int sub=S;
3 do{
4 //對某集合的子集合的處理
5 sub=(sub-1)&S;
6 }while(sub!=S);
7 }
8 void k_sub_set(int k,int n){
9 int comb=(1<<k)-1,S=1<<n;
10 while(comb<S){
11 //對大小為k的子集合的處理
12 int x=comb&-comb,y=comb+x;
13 comb=((comb&~y)/x>>1)|y;
14 }
15 }

```

4.3 matrix exponential

```

1 void exp(LL m[2][2], LL x){
2 LL c[2][2] = {{1,1},{1,0}},n[2][2];
3 n[0][0] = m[0][0]*c[0][0] + m[0][1]*c
4 [1][0];
5 n[0][1] = m[0][0]*c[0][1] + m[0][1]*c
6 [1][1];
7 n[1][0] = m[1][0]*c[0][0] + m[1][1]*c
8 [1][0];
9 n[1][1] = m[1][0]*c[0][1] + m[1][1]*c
10 [1][1];
11 if(x != 1)
12 exp(n,x-1);
13 else
14 cout << n[0][0];
15 }
16 int main(){
17 LL u[2][2] = {{1,1},{1,0}},n;
18 cin >> n;
19 cout << "gOxRC²" << n+2 << "gμ-°";
20 exp(u,n);
21 }

```

4.4 SpeedExpo

```

1 LL expo(LL a,LL b,LL p){
2 if(b == 0) return 1;
3 if(b & 1) return (expo(a,b-1,p)*a)%p;
4 //b is odd
5 LL temp = expo(a,b/2,p);
6 return (temp*temp)%p;
7 }

```

4.5 外星模運算

```

1 //a[0]^a[1]^a[2]^...
2 #define maxn 1000000
3 int euler[maxn+5];
4 bool is_prime[maxn+5];
5 void init_euler(){
6 is_prime[1]=1; //一不是質數
7 for(int i=1;i<=maxn;i++)euler[i]=i;
8 for(int i=2;i<=maxn;i++){
9 if(!is_prime[i]){ //是質數
10 euler[i]--;
11 for(int j=i<1;j<=maxn;j+=i){

```

```

12 is_prime[j]=1;
13 euler[j]=euler[j]/i*(i-1);
14 }
15 }
16 }
17 LL pow(LL a,LL b,LL mod){ //a^b mod
18 LL ans=1;
19 for(;b;a=a%mod,b>>=1)
20 if(b&1)ans=ans*a%mod;
21 return ans;
22 }
23 bool isless(LL *a,int n,int k){
24 if(*a==1)return k>1;
25 if(--n==0)return *a<k;
26 int next=0;
27 for(LL b=1;b<k;next)
28 b*=*a;
29 return isless(a+1,n,next);
30 }
31 LL high_pow(LL *a,int n,LL mod){
32 if(*a==1||--n==0)return *a%mod;
33 int k=0,r=euler[mod];
34 for(LL tma=1;tma!=pow(*a,k+r,mod);++k)
35 tma=tma*(a)%mod;
36 if(isless(a+1,n,k))return pow(*a,
37 high_pow(a+1,n,k),mod);
38 int tmd=high_pow(a+1,n,r), t=(tmd-k+r)%
39 r;
40 return pow(*a,k+t,mod);
41 }
42 LL a[1000005];
43 int t,mod;
44 int main(){
45 init_euler();
46 scanf("%d",&t);
47 #define n 4
48 while(t--){
49 for(int i=0;i<n;++i)scanf("%lld",&a[i]);
50 scanf("%d",&mod);
51 printf("%lld\n",high_pow(a,n,mod));
52 }
53 return 0;
54 }

```

4.6 大數取模

```

1 LL exp(LL x,LL y,LL p){
2 if(y == 0) return 1;
3 if(y & 1) return (exp(x,y-1,p)*x) % p
4 //y is odd
5 else{
6 LL temp = exp(x,y/2,p);
7 return (temp*temp) % p;
8 }
9 LL calcmo(LL index,LL p){
10 if(index == 0) return base[index] - '0'
11 ;
12 LL single = calcmo(index-1,p)*10;
13 return (single%p + base[index] - '0')%p
14 ;
15 }

```

4.7 模逆元

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 ;cin.tie(0)
4 typedef long long LL;
5 using namespace std;
6 LL mod_inverse_by_speed_exp(LL x,LL y,LL
7 p){
8 if(y == 0) return 1;
9 if(y % 2) return (
10 mod_inverse_by_speed_exp(x,y-1,p)
11 *x)%p;
12 else{
13 LL temp =
14 mod_inverse_by_speed_exp(x,y
15 /2,p);
16 return (temp*temp)%p;
17 }
18 }
19 int main(){
20 good;
21 LL n,i,p,x;
22 cin >> n >> p;
23 for(i = 0; i < n; i++){

```



```

19     cin >> x;
20     cout << mod_inverse_by_speed_exp(
21         x,p-2,p) << ' ';
22     }
23     return 0;

```

```

81     }
82     now*=tmp[i];
83     for(int j=0;j<len;++j)
84         v.push_back(v[j]*now);
85     }
86 }

```

```

36
37 int main(){
38     good;
39     //freopen("file name", "r", stdin);
40     //input redirection
41     LL i,n,m;
42     cin >> n >> m;
43     for(i = 0; i < m; i++){
44         LL x,y,weight;
45         cin >> x >> y >> weight;
46         adjacent.push_back({x,y,weight});
47     }
48     memset(fa,-1,sizeof(fa)); //unvisited
49     sort(adjacent.begin(),adjacent.end(),
50         cmp); //sort by weight
51     LL cost = 0, now_edge = 0;
52     for(EDGE e : adjacent){
53         if(merge(e.u,e.v)){ //connect edge
54             cost += e.w;
55             now_edge++;
56         }
57     }
58     if(now_edge < n-1) //not a MST
59         cout << -1 << endl;
60     else
61         cout << cost << endl;
62     return 0;
63 }

```

4.8 質因數分解

5 String

5.1 manacher(最小回文字串)

```

1 LL func(const LL n,const LL mod,const int
2     c) {
3     return (LLmul(n,n,mod)+c+mod)%mod;
4 }
5 LL pollorroho(const LL n, const int c) { //
6     循環節長度
7     LL a=1, b=1;
8     a=func(a,n,c)%n;
9     b=func(b,n,c)%n; b=func(b,n,c)%n;
10    while(gcd(abs(a-b),n)==1) {
11        a=func(a,n,c)%n;
12        b=func(b,n,c)%n; b=func(b,n,c)%n;
13    }
14    return gcd(abs(a-b),n);
15 }
16 void prefactor(LL &n, vector<LL> &v) {
17     for(int i=0;i<12;++i) {
18         while(n%prime[i]==0) {
19             v.push_back(prime[i]);
20             n/=prime[i];
21         }
22     }
23 }
24 void smallfactor(LL n, vector<LL> &v) {
25     if(n<MAXPRIME) {
26         while(isp[(int)n]) {
27             v.push_back((int)n);
28             n/=isp[(int)n];
29         }
30     }
31     v.push_back(n);
32 } else {
33     for(int i=0;i<primecnt&&prime[i]*
34         prime[i]<=n;++i) {
35         while(n%prime[i]==0) {
36             v.push_back(prime[i]);
37             n/=prime[i];
38         }
39     }
40     if(n!=1) v.push_back(n);
41 }
42 }
43 void comfactor(const LL &n, vector<LL> &v) {
44     if(n<1e9) {
45         smallfactor(n,v);
46         return;
47     }
48     if(Isprime(n)) {
49         v.push_back(n);
50         return;
51     }
52     LL d;
53     for(int c=3; c<=n; c++) {
54         d = pollorroho(n,c);
55         if(d!=n) break;
56     }
57     comfactor(d,v);
58     comfactor(n/d,v);
59 }
60 }
61 void Factor(const LL &x, vector<LL> &v) {
62     LL n = x;
63     if(n==1) { puts("Factor 1"); return; }
64     prefactor(n,v);
65     if(n==1) return;
66     comfactor(n,v);
67     sort(v.begin(),v.end());
68 }
69 void AllFactor(const LL &n,vector<LL> &v) {
70     vector<LL> tmp;
71     Factor(n,tmp);
72     v.clear();
73     v.push_back(1);
74     int len;
75     LL now=1;
76     for(int i=0;i<tmp.size();++i) {
77         if(i==0 || tmp[i]!=tmp[i-1]) {
78             len = v.size();
79             now = 1;

```

```

1 //原字串: asdsasdsa
2 //要先把字串變成這樣: @#a#s#d#s#a#s#d#s#a
3 #
4 void manacher(char *s,int len,int *z){
5     int l=0,r=0;
6     for(int i=1;i<len;++i){
7         z[i]=r>i?min(z[2*i-l],r-i):1;
8         while(s[i+z[i]]==s[i-z[i]])++z[i];
9         if(z[i]+i>r)r=z[i]+i,l=i;
10    } //ans = max(z)-1
11 }

```

6 Tree Problem

6.3 LCA

6.1 findLCA

```

1 LL findLCA(LL u,LL v){
2     while(depth[u] > depth[v])
3         u = father[u];
4     while(depth[v] > depth[u])
5         v = father[v];
6     while(u != v){
7         u = father[u];
8         v = father[v];
9     }
10    return u; //or return v
11 }

```

```

1 const int MAXN=100000; // 1-base
2 const int MLG=17; //Log2(MAXN)+1;
3 int pa[MLG+2][MAXN+5];
4 int dep[MAXN+5];
5 vector<int> G[MAXN+5];
6 void dfs(int x,int p=0){ //dfs(root);
7     pa[0][x]=p;
8     for(int i=0;i<=MLG;++i)
9         pa[i+1][x]=pa[i][pa[i][x]];
10    for(auto &i:G[x]){
11        if(i==p) continue;
12        dep[i]=dep[x]+1;
13        dfs(i,x);
14    }
15 }
16 inline int jump(int x,int d){
17     for(int i=0;i<=MLG;++i)
18         if((d>>i)&1) x=pa[i][x];
19     return x;
20 }
21 inline int find_lca(int a,int b){
22     if(dep[a]>dep[b]) swap(a,b);
23     b=jump(b,dep[b]-dep[a]);
24     if(a==b) return a;
25     for(int i=MLG;i>=0;--i){
26         if(pa[i][a]!=pa[i][b]){
27             a=pa[i][a];
28             b=pa[i][b];
29         }
30     }
31     return pa[0][a];
32 }

```

6.2 kruskal(MST)

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 ;cin.tie(0)
4 #define RSIZE 10002
5 #define pll pair<LL,LL>
6 typedef long long LL;
7 using namespace std;
8 struct EDGE{
9     LL u,v,w;
10 };
11 vector<EDGE> adjacent; //out neighbor,
12 weight
13 LL fa[RSIZE];
14 bool cmp(EDGE &a,EDGE &b){ //sort by
15     weight
16     return a.w < b.w;
17 }
18 LL sfind(LL now){ //find root, root's
19     father=set size
20     if(fa[now] < 0)
21         return now;
22     return fa[now] = sfind(fa[now]);
23 }
24 bool merge(LL u,LL v){ //find two root,
25     comparing size(by root's father)
26     LL set1 = sfind(u), set2 = sfind(v);
27     if(set1 == set2) return false; //same
28     root-> no need to merge
29     if(fa[set1] < fa[set2]){ //set1 is
30         larger
31         fa[set1] += fa[set2];
32         fa[set2] = set1;
33     }
34     else{
35         fa[set2] += fa[set1];
36         fa[set1] = set2;
37     }
38     return true;
39 }

```

6.4 Prim(MST)

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 ;cin.tie(0)
4 #define RSIZE 502
5 #define oo 1000000001 //1e9+1
6 typedef long long LL;
7 using namespace std;
8 vector<pair<LL,LL>> adjacent[RSIZE]; //out
9 neighbor, weight of edge
10 LL dis[RSIZE], fa[RSIZE]; //dis for weight
11 of two vertexes
12 bool visit[RSIZE] = {false};
13 int main(){
14     good;
15     //freopen("file name", "r", stdin);
16     //input redirection
17     LL i,n,m;
18     cin >> n >> m;
19     for(i = 0; i < m; i++){
20         LL x,y,w;
21         cin >> x >> y >> w;
22         adjacent[x].push_back({y,w});
23         adjacent[y].push_back({x,w});
24     }

```

```

22 }
23 //initial
24 LL start = 0;
25 memset(dis,0,sizeof(dis));
26 memset(fa,-1,sizeof(fa));
27 priority_queue<pair<LL,LL>> PQ; //dis
    [,vertex
28 PQ.push({dis[start] = 0,start});
29 //prim
30 while (!PQ.empty()){
31     auto pt = PQ.top();
32     PQ.pop();
33     LL v = pt.second;
34     if(visit[v]) continue;
35     visit[v] = true;
36     for(auto it : adjacent[v]){
37         LL neighbor = it.first,w = it.
            second;
38         if(visit[neighbor]) continue;
39         if(w < dis[neighbor]){ //new
            edge is shorter
40             dis[neighbor] = w;
41             fa[neighbor] = v;
42             PQ.push({-dis[neighbor],
                neighbor});
43         }
44     }
45 }
46 LL cost = 0,cnt = 0;
47 //count cost and check if MST exists
48 for(i = 0; i < n; i++){
49     if(dis[i] < oo)
50         cost += dis[i];
51     else
52         cnt++;
53 }
54 if(cnt)
55     cout << -1 << endl;
56 else
57     cout << cost << endl;
58 return 0;
59 }

```

6.5 segment tree

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
    ;cin.tie(0)
3 #define RSIZE 100000
4 #define pll pair<LL,LL>
5 #define lc 2*index //c means child
6 #define rc 2*index+1
7 typedef long long LL;
8 using namespace std;
9
10 LL tree[4*RSIZE]; //saving range maximum
11 LL lazy[4*RSIZE] = {0};
12 LL num[RSIZE],cnt = 1;
13
14 //using range maximum as example
15 void build(LL L,LL R,LL index){
16     LL temp = index;
17     if(L == R){
18         tree[index] = num[cnt];
19         cnt++;
20         return;
21     }
22     LL M = (L+R)/2;
23     build(L,M,lc);
24     build(M+1,R,rc);
25     tree[index] = max(tree[lc],tree[rc]);
26 }
27 //single point modify
28 void modify(LL x,LL v,LL L,LL R,LL index)
    {
29     if(L == R){
30         tree[index] = v;
31         return;
32     }
33     LL M = (L+R)/2;
34     if(x <= M) //left side
35         modify(x,v,L,M,lc);
36     else
37         modify(x,v,M+1,R,rc);
38     tree[index] = max(tree[lc],tree[rc]);
39 }
40 //a range including index has to add tag
41 void addtag(LL tag,LL index){
42     tree[index] += tag;
43     lazy[index] += tag;
44 }
45 //transferring tag to child
46 void push(LL index){
47     addtag(lazy[index],lc);

```

```

48     addtag(lazy[index],rc);
49     lazy[index] = 0; //tag is transfered
        to child
50 }
51 //Lower variables are queried range,UPPER
    ones are full range
52 LL query(LL l,LL r,LL L,LL R,LL index){
53     if(l <= L && R <= r) return tree[
        index];
54     push(index); //if use single point
        modify,no need
55     LL M = (L+R)/2;
56     if(r <= M) //answer in the left side,
        don't need to query right side
57         return query(l,r,L,M,lc);
58     else if(l > M) //in right side
59         return query(l,r,M+1,R,rc);
60     else //answer cross both side
61         return max(query(l,r,L,M,lc),
            query(l,r,M+1,R,rc)); //
            choose better one
62 }
63 void multi_modify(LL l,LL r,LL v,LL L,LL
    R,LL index){
64     if(l <= L && R <= r){
65         addtag(v,index);
66         return;
67     }
68     push(index);
69     LL M = (L+R)/2;
70     if(r <= M) multi_modify(l,r,v,L,M,lc)
        ;
71     else if(l > M) multi_modify(l,r,v,M
        +1,R,rc);
72     else{
73         multi_modify(l,r,v,L,M,lc);
74         multi_modify(l,r,v,M+1,R,rc);
75     }
76     tree[index] = max(tree[lc],tree[rc]);
77 }
78
79 int main(){
80     good;
81     LL k,n;
82     //build(1,n,1)at first,can use query(
        L,r,1,n,1).
83     return 0;
84 }

```

7 default

7.1 8 queen

```

1 LL nqueen(LL n){
2     int p[17],total = 0;
3     for(int i = 0; i < n; i++)
4         p[i] = i;
5     do{
6         bool valid = true;
7         for(int i = 0; i < n; i++){
8             for(int j = i+1; j < n; j++){
9                 if(abs(p[i]-p[j]) == j-i)
                { //same diagonal
10                     valid = false;
11                     break;
12                 }
13             }
14         }
15         if(valid) total++;
16         while (next_permutation(p,p+n));
17         return total;
18 }

```

7.2 debug

```

1 #ifdef DEBUG
2 #define dbg(...) {\
3     fprintf(stderr,"%s - %d : (%s) = ",
        __PRETTY_FUNCTION__,__LINE__,#
        __VA_ARGS__); \
4     _DO(__VA_ARGS__); \
5 }
6 template<typename I> void _DO(I&&x){cerr
    <<<<endl;}
7 template<typename I,typename...T> void
    _DO(I&&x,T&&...tail){cerr<<x<<" ";
    _DO(tail...);}
8 #else
9 #define dbg(...)

```

```

10 #endif
11
12 //Magic
13 #pragma GCC optimize "Ofast"
14 //stack resize,change esp to rsp if 64-
    bit system
15 asm("mov %0,%esp\n" :: "g"(mem+10000000))
    ;
16 -Wl,--stack,214748364 -trigraphs
17 #pragma comment(linker, "/STACK
    :1024000000,1024000000")
18 //Linux stack resize
19 #include<sys/resource.h>
20 void increase_stack(){
21     const rlim_t ks=64*1024*1024;
22     struct rlimit rl;
23     int res=getrlimit(RLIMIT_STACK,&rl);
24     if(!res&&rl.rlim_cur<ks){
25         rl.rlim_cur=ks;
26         res=setrlimit(RLIMIT_STACK,&rl);
27     }

```

7.3 IncStack

```

1 //Magic
2 #pragma GCC optimize "Ofast"
3 //stack resize,change esp to rsp if 64-
    bit system
4 asm("mov %0,%esp\n" :: "g"(mem+10000000))
    ;
5 -Wl,--stack,214748364 -trigraphs
6 #pragma comment(linker, "/STACK
    :1024000000,1024000000")
7 //Linux stack resize
8 #include<sys/resource.h>
9 void increase_stack(){
10     const rlim_t ks=64*1024*1024;
11     struct rlimit rl;
12     int res=getrlimit(RLIMIT_STACK,&rl);
13     if(!res&&rl.rlim_cur<ks){
14         rl.rlim_cur=ks;
15         res=setrlimit(RLIMIT_STACK,&rl);
16     }
17 }

```

7.4 input

```

1 inline int read(){
2     int x=0; bool f=0; char c=getchar();
3     while(ch<'0' || '9'<ch) f|=ch=='-' ,ch=
        getchar();
4     while('0'<=ch&&ch<='9') x=x*10-'0'+ch,ch
        =getchar();
5     return f?-x:x;
6 }
7 // #!/bin/bash
8 // g++ -std=c++11 -O2 -Wall -Wextra -Wno-
    unused-result -DDEBUG $1 && ./a.out
9 // -fsanitize=address -fsanitize=
    undefined -fsanitize=return

```

7.5 randomize

```

1 map<LL,LL> discret;
2 for(i = 0; i < n; i++){
3     cin >> a[i];
4     discret[a[i]] = 0;
5 }
6 LL index = 0;
7 for(auto &it : discret)
8     it.second = index++;

```

7.6 sweepline

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
    ;cin.tie(0)
4 typedef long long LL;
5 using namespace std;
6
7 struct Seg{
8     LL left,right;
9 };
10
11 bool cmp(Seg &a,Seg &b){
12     return a.left < b.left;
13 }
14
15
16 int main(){
17     good;
18     //freopen("P_2_1_5.in", "r", stdin)
        ; //input redirection
19     LL n;
20     cin >> n;
21     Seg line[n];
22     for(LL i = 0; i < n; i++)
23         cin >> line[i].left >> line[i].
            right;
24     sort(line,line+n,cmp);
25     Seg last = line[0];
26     LL total = 0;
27     for(LL i = 1; i < n; i++){

```

```

28         if(line[i].left > last.right){
29             total += last.right - last.
                left;
30             last = line[i];
31             continue;
32         }
33         last.right = max(last.right, line[
            i].right); //merge last and
                line[i]
34     }
35     total += last.right - last.left;
36     cout << total;
37     return 0;
38 }

```

7.7 模板

```

1 #include <bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(0)
3 #define RSIZE 101
4 #define pll pair<LL,LL>
5 #define lc 2*index
6 #define rc 2*index+1
7 typedef long long LL;
8 using namespace std;
9
10 int main(){
11     good;
12     //freopen("file name", "r", stdin);
        //input redirection
13
14     return 0;
15 }

```

8 other

8.1 1D0D dp

```

1 #include <bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(0)
3 typedef long long LL;
4 using namespace std;
5
6 int main(){
7     good;
8     //freopen("file name", "r", stdin);
        //input redirection
9
10     LL i, n;
11     LL money[100002], dp[100002] = {0}; //dp
        for last i day max profit
12     cin >> n;
13     for(i = 1; i <= n; i++)
14         cin >> money[i];
15     dp[0] = 0, dp[1] = money[1], dp[2] =
        max(money[1], money[2]);
16     for(i = 3; i <= n; i++)
17         dp[i] = max(dp[i-2]+money[i], dp[i
            -1]); //choose, discard
18     cout << dp[i-1];
19     return 0;
20 }

```

8.2 mergesort

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(0)
4 #define NORMALSIZE 8
5 typedef long long LL;
6
7 void merge(LL *, LL, LL, LL); //prototype
8 void mergesort(LL *, LL, LL);
9
10 void mergesort(LL* arr, LL l, LL r){
11     LL m;
12     if(l < r){
13         m = (l+r)/2;
14         mergesort(arr, l, m); //sort left
            subarray
15         mergesort(arr, m+1, r); //sort right
            subarray
16         merge(arr, l, m, r);

```

```

17     }
18     return;
19 }
20 void merge(LL* arr, LL l, LL m, LL r){
21     LL left, right, tmp[NORMALSIZE], i;
22     left = l; // left subarr.begin
23     right = m+1; //right subarr.begin
24     i = l;
25     while ((left <= m) && (right <= r)){
26         //merging left subarr and right
            subarr
27         if(arr[left] < arr[right]){ //
            left subarr is smaller
28             tmp[i] = arr[left];
29             i++, left++;
30         }
31         else{ //the other situation
32             tmp[i] = arr[right];
33             i++, right++;
34         }
35     }
36     while(left <= m){ // right subarr is
        fully sorted
37         tmp[i] = arr[left];
38         i++, left++;
39     }
40     while(right <= r){ // left subarr is
        fully sorted
41         tmp[i] = arr[right];
42         i++, right++;
43     }
44     for(i = l; i <= r; i++)
45         arr[i] = tmp[i];

```

8.3 WhatDay

```

1 int whatday(int y, int m, int d){
2     if(m <= 2) m += 12, --y;
3     if(y < 1752 || |y == 1752 && m < 9 || y == 1752 && m
        == 9 && d < 3)
4         return (d + 2 * m + 3 * (m + 1) / 5 + y / 4 + 5) % 7;
5     return (d + 2 * m + 3 * (m + 1) / 5 + y / 4 - y / 100 + y
        / 400) % 7;
6 }

```

9 zformula

9.1 formula

9.1.1 Pick 公式

給定頂點坐標均是整點的簡單多邊形，面積 = 內部格點數 + 邊上格點數/2 - 1

9.1.2 圖論

- 對於平面圖， $F = E - V + C + 1$ ， C 是連通分量數
- 對於平面圖， $E \leq 3V - 6$
- 對於連通圖 G ，最大獨立點集的大小設為 $I(G)$ ，最大匹配大小設為 $M(G)$ ，最小點覆蓋設為 $C_v(G)$ ，最小邊覆蓋設為 $C_e(G)$ 。對於任意連通圖：

- $I(G) + C_v(G) = |V|$
- $M(G) + C_e(G) = |V|$

- 對於連通二分圖：

- $I(G) = C_v(G)$
- $M(G) = C_e(G)$

- 最大權閉合圖：

- $C(u, v) = \infty, (u, v) \in E$
- $C(S, v) = W_v, W_v > 0$
- $C(v, T) = -W_v, W_v < 0$
- $ans = \sum_{W_v > 0} W_v - flow(S, T)$

- 最大密度子圖：

- 求 $max \left(\frac{w_e + W_v}{|V'|} \right), e \in E', v \in V'$
- $U = \sum_{v \in V} 2W_v + \sum_{e \in E} W_e$
- $C(u, v) = W_{(u,v)}, (u, v) \in E$ ，雙向邊
- $C(S, v) = U, v \in V$
- $D_u = \sum_{(u,v) \in E} W_{(u,v)}$
- $C(v, T) = U + 2g - D_v - 2W_v, v \in V$

- 二分搜 g ：
 $l = 0, r = U, eps = 1/n^2$
 if $((U \times |V| - flow(S, T)) / 2 > 0)$ $l = mid$
 else $r = mid$
 - $ans = min_cut(S, T)$
 - $|E| = 0$ 要特殊判斷
7. 弦圖：

- 點數大於 3 的環都要有一條弦
- 完美消除序列從後往前依次給每個點染色，給每個點染上可以染的最小顏色
- 最大團大小 = 色數
- 最大獨立集：完美消除序列從前往後能選就選
- 最小團覆蓋：最大獨立集的點和他延伸的邊構成
- 區間圖是弦圖
- 區間圖的完美消除序列：將區間按造又端點由小到大排序
- 區間圖染色：用線段樹做

9.1.3 dinic 特殊圖複雜度

- 單位流： $O \left(\min \left(V^{3/2}, E^{1/2} \right) E \right)$
- 二分圖： $O \left(V^{1/2} E \right)$

9.1.4 0-1 分數規劃

$x_i = \{0, 1\} \cdot x_i$ 可能會有其他限制，求 $max \left(\frac{\sum B_i x_i}{\sum C_i x_i} \right)$

- $D(i, g) = B_i - g \times C_i$
- $f(g) = \sum D(i, g) x_i$
- $f(g) = 0$ 時 g 為最佳解， $f(g) < 0$ 沒有意義
- 因為 $f(g)$ 單調可以二分搜 g
- 或用 Dinkelbach 通常比較快

```

1 binary_search(){
2     while(r-l>eps){
3         g=(l+r)/2;
4         for(i:所有元素)D[i]=B[i]-g*C[i]; //D(i
            ,g)
5         找出一組合法x[i]使f(g)最大;
6         if(f(g)>0) l=g;
7         else r=g;
8     }
9     Ans = r;
10 }
11 Dinkelbach(){
12     g=任意狀態 (通常設為0);
13     do{
14         Ans=g;
15         for(i:所有元素)D[i]=B[i]-g*C[i]; //D(i
            ,g)
16         找出一組合法x[i]使f(g)最大;
17         p=0, q=0;
18         for(i:所有元素)
19             if(x[i]p+=B[i], q+=C[i];
20         g=p/q; //更新解，注意q=0的情況
21     }while(abs(Ans-g)>EPS);
22     return Ans;
23 }

```

9.1.5 學長公式

- $\sum_{d|n} \phi(n) = n$
- $g(n) = \sum_{d|n} f(d) \Rightarrow f(n) = \sum_{d|n} \mu(d) \times g(n/d)$
- Harmonic series $H_n = \ln(n) + \gamma + 1/(2n) - 1/(12n^2) + 1/(120n^4)$
- $\gamma = 0.57721566490153286060651209008240243104215$
- 格雷碼 $= n \oplus (n >> 1)$
- $SG(A + B) = SG(A) \oplus SG(B)$
- 選轉矩陣 $M(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$

9.1.6 基本數論

- $\sum_{d|n} \mu(n) = [n == 1]$
- $g(m) = \sum_{d|m} f(d) \Leftrightarrow f(m) = \sum_{d|m} \mu(d) \times g(m/d)$
- $\sum_{i=1}^n \sum_{j=1}^m$ 互質數量 $= \sum \mu(d) \lfloor \frac{n}{d} \rfloor \lfloor \frac{m}{d} \rfloor$
- $\sum_{i=1}^n \sum_{j=1}^m lcm(i, j) = n \sum_{d|n} d \times \phi(d)$

9.1.7 排組公式

1. k 卡特蘭 $\frac{C_n^{kn}}{n(k-1)+1} \cdot C_m^n = \frac{n!}{m!(n-m)!}$
2. $H(n, m) \cong x_1 + x_2 \dots + x_n = k, num = C_k^{n+k-1}$
3. Stirling number of $2^{nd}, n$ 人分 k 組方法數目
 - (a) $S(0, 0) = S(n, n) = 1$
 - (b) $S(n, 0) = 0$
 - (c) $S(n, k) = kS(n-1, k) + S(n-1, k-1)$
4. Bell number, n 人分任意多組方法數目
 - (a) $B_0 = 1$
 - (b) $B_n = \sum_{i=0}^n S(n, i)$
 - (c) $B_{n+1} = \sum_{k=0}^n C_k^n B_k$
 - (d) $B_{p+n} \equiv B_n + B_{n+1} \pmod{p}$, p is prime
 - (e) $B_{p^m+n} \equiv mB_n + B_{n+1} \pmod{p}$, p is prime
 - (f) From $B_0 : 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975$
5. Derangement, 錯排, 沒有人在自己位置上
 - (a) $D_n = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \dots + (-1)^n \frac{1}{n!})$
 - (b) $D_n = (n-1)(D_{n-1} + D_{n-2}), D_0 = 1, D_1 = 0$
 - (c) From $D_0 : 1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496$

6. Binomial Equality

- (a) $\sum_k \binom{r}{m+k} \binom{s}{n-k} = \binom{r+s}{m+n}$
- (b) $\sum_k \binom{l}{m+k} \binom{s}{n+k} = \binom{l+s}{l+m+n}$
- (c) $\sum_k \binom{l}{m+k} \binom{s+k}{n} (-1)^k = (-1)^{l+m} \binom{s-m}{n-l}$
- (d) $\sum_{k \leq l} \binom{l-k}{m} \binom{s}{k-n} (-1)^k = (-1)^{l+m} \binom{s-m-1}{l-n-m}$
- (e) $\sum_{0 \leq k \leq l} \binom{l-k}{m} \binom{q+k}{n} = \binom{l+q+1}{m+n+1}$
- (f) $\binom{r}{k} = (-1)^k \binom{k-r-1}{k}$
- (g) $\binom{r}{m} \binom{m}{k} = \binom{r}{k} \binom{r-k}{m-k}$
- (h) $\sum_{k \leq n} \binom{r+k}{k} = \binom{r+n+1}{n}$
- (i) $\sum_{0 \leq k \leq n} \binom{k}{m} = \binom{n+1}{m+1}$
- (j) $\sum_{k \leq m} \binom{m+r}{k} x^k y^k = \sum_{k \leq m} \binom{-r}{k} (-x)^k (x+y)^{m-k}$

9.1.10 Count on a tree

1. Rooted tree: $s_{n+1} = \frac{1}{n} \sum_{i=1}^n (i \times a_i \times \sum_{j=1}^{\lfloor n/i \rfloor} a_{n+1-i \times j})$
2. Unrooted tree:
 - (a) Odd: $a_n - \sum_{i=1}^{n/2} a_i a_{n-i}$
 - (b) Even: $Odd + \frac{1}{2} a_{n/2} (a_{n/2} + 1)$
3. Spanning Tree
 - (a) 完全圖 $n^n - 2$
 - (b) 一般圖 (Kirchhoff's theorem) $M[i][i] = \text{degree}(V_i), M[i][j] = -1, \text{if have } E(i, j), 0 \text{ if no edge. delete any one row and col in } A, \text{ans} = \det(A)$

9.1.8 幕次, 幕次和

1. $a^{b \% \varphi P} = a^{b \% \varphi(p) + \varphi(p)}, b \geq \varphi(p)$
2. $1^3 + 2^3 + 3^3 + \dots + n^3 = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4}$
3. $1^4 + 2^4 + 3^4 + \dots + n^4 = \frac{n^5}{5} + \frac{n^4}{2} + \frac{n^3}{3} - \frac{n}{30}$
4. $1^5 + 2^5 + 3^5 + \dots + n^5 = \frac{n^6}{6} + \frac{n^5}{2} + \frac{5n^4}{12} - \frac{n^2}{12}$
5. $0^k + 1^k + 2^k + \dots + n^k = P(k), P(k) = \frac{(n+1)^{k+1} - \sum_{i=0}^{k-1} C_i^{k+1} P(i)}{k+1}, P(0) = n+1$
6. $\sum_{k=0}^{m-1} k^n = \frac{1}{n+1} \sum_{k=0}^n C_k^{n+1} B_k m^{n+1-k}$
7. $\sum_{j=0}^m C_j^{m+1} B_j = 0, B_0 = 1$
8. 除了 $B_1 = -1/2$ · 剩下的奇數項都是 0
9. $B_2 = 1/6, B_4 = -1/30, B_6 = 1/42, B_8 = -1/30, B_{10} = 5/66, B_{12} = -691/2730, B_{14} = 7/6, B_{16} = -3617/510, B_{18} = 43867/798, B_{20} = -174611/330,$

9.1.9 Burnside's lemma

1. $|X/G| = \frac{1}{|G|} \sum_{g \in G} |X^g|$
2. $X^g = t^{c(g)}$
3. G 表示有幾種轉法 · X^g 表示在那種轉法下 · 有幾種是會保持對稱的 · t 是顏色數 · $c(g)$ 是循環節不動的面數 ·
4. 正立方體塗三顏色 · 轉 0 有 3^6 個元素不變 · 轉 90 有 6 種 · 每種有 3^3 不變 · 180 有 3×3^4 · 120(角) 有 8×3^2 · 180(邊) 有 6×3^3 · 全部 $\frac{1}{24} (3^6 + 6 \times 3^3 + 3 \times 3^4 + 8 \times 3^2 + 6 \times 3^3) = 57$

Codebook - ss

Contents

1 Computational Geometry	1	3.3 DFS	2	7 default	6
1.1 SmallestCircle	1	3.4 dijkstra	3	7.1 8 queen	6
1.2 最近點對	1	3.5 topology sort	3	7.2 debug	6
2 Data Structure	1	3.6 union and find	3	7.3 IncStack	6
2.1 01 背包	1	4 Number Theory	3	7.4 input	6
2.2 array simulate linked list .	1	4.1 basic	3	7.5 randomize	6
2.3 binary search	1	4.2 bit set	4	7.6 sweepline	6
2.4 discretization	1	4.3 matrix exponential	4	7.7 模板	7
2.5 half enumeration	1	4.4 SpeedExpo	4	8 other	7
2.6 LCS	2	4.5 外星模運算	4	8.1 1D0D dp	7
2.7 LIS	2	4.6 大數取模	4	8.2 mergesort	7
2.8 skew heap	2	4.7 模逆元	4	8.3 WhatDay	7
2.9 sliding window	2	4.8 質因數分解	5	9 zformula	7
2.10 undo disjoint set	2	5 String	5	9.1 formula	7
3 Graph	2	5.1 manacher(最小回文字串)	5	9.1.1 Pick 公式	7
3.1 bellman ford	2	6 Tree Problem	5	9.1.2 圖論	7
3.2 BFS	2	6.1 findLCA	5	9.1.3 dinic 特殊圖複雜度	7
		6.2 kruskal(MST)	5	9.1.4 0-1 分數規劃 . . .	7
		6.3 LCA	5	9.1.5 學長公式	7
		6.4 Prim(MST)	5	9.1.6 基本數論	7
		6.5 segment tree	6	9.1.7 排組公式	8
				9.1.8 冪次, 冪次和 . . .	8
				9.1.9 Burnside's lemma .	8
				9.1.10 Count on a tree . .	8

Codebook - ss

C++ Resource Test

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 namespace system_test {
5
6     const size_t KB = 1024;
7     const size_t MB = KB * 1024;
8     const size_t GB = MB * 1024;
9
10    size_t block_size, bound;
11    void stack_size_dfs(size_t depth = 1) {
12        if (depth >= bound)
13            return;
14        int8_t ptr[block_size]; // 若無法編譯將
15                                // block_size 改成常數
16        memset(ptr, 'a', block_size);
17        cout << depth << endl;
18        stack_size_dfs(depth + 1);
19    }
20
21    void stack_size_and_runtime_error(size_t
22        block_size, size_t bound = 1024) {
23        system_test::block_size = block_size;
24        system_test::bound = bound;
25        stack_size_dfs();
26    }
27 }
```

```
24 }
25
26 double speed(int iter_num) {
27     const int block_size = 1024;
28     volatile int A[block_size];
29     auto begin = chrono::
30         high_resolution_clock::now();
31     while (iter_num--)
32         for (int j = 0; j < block_size; ++j)
33             A[j] += j;
34     auto end = chrono::
35         high_resolution_clock::now();
36     chrono::duration<double> diff = end -
37         begin;
38     return diff.count();
39 }
40
41 void runtime_error_1() {
42     // Segmentation fault
43     int *ptr = nullptr;
44     *(ptr + 7122) = 7122;
45 }
46
47 void runtime_error_2() {
48     // Segmentation fault
49     int *ptr = (int *)memset;
50     *ptr = 7122;
51 }
52
53 void runtime_error_3() {
54     // munmap_chunk(): invalid pointer
55     int *ptr = (int *)memset;
56     delete ptr;
57 }
```

```
56 void runtime_error_4() {
57     // free(): invalid pointer
58     int *ptr = new int[7122];
59     ptr += 1;
60     delete[] ptr;
61 }
62
63 void runtime_error_5() {
64     // maybe illegal instruction
65     int a = 7122, b = 0;
66     cout << (a / b) << endl;
67 }
68
69 void runtime_error_6() {
70     // floating point exception
71     volatile int a = 7122, b = 0;
72     cout << (a / b) << endl;
73 }
74
75 void runtime_error_7() {
76     // call to abort.
77     assert(false);
78 }
79
80 } // namespace system_test
81
82 #include <sys/resource.h>
83 void print_stack_limit() { // only work
84     in Linux
85     struct rlimit l;
86     getrlimit(RLIMIT_STACK, &l);
87     cout << "stack_size = " << l.rlim_cur
88         << " byte" << endl;
89 }
```