

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     }
16     for(int j=0;j<i;j++){
17         if((p[j]-c).abs2()>r2)
18             c.x=(p[i].x+p[j].x)/2;
19             c.y=(p[i].y+p[j].y)/2;
20             r2=(p[j]-c).abs2();
21     }
22     for(int k=0;k<j;k++){
23         if((p[k]-c).abs2()>r2)
24             c=circumcenter(p[i],p[j],p[k]);
25             r2=(p[i]-c).abs2();
26     }
27 }

```

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            {
14                T dx=u->x-v->x, dy=u->y-v->y;
15                if(dy*dy>=d) break;
16                d=min(d,dx*dx+dy*dy);
17            }
18        b.push_back(*u);
19    }
20    return d;
21 }
22 T closest_pair(vector<point<T>> &v){
23     sort(v.begin(),v.end(),xcmp);
24     return closest_pair(v.begin(),v.end());
25 }

```

2 Data Structure

2.1 01 背包

```

1 LL dp[101][10001] = {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 }
6 LL r=0;
7 for (auto it=S.begin(); it!=S.end();
8     ++it) //traversal and set rank
9     it->second = r++;
10 // replace number with rank
11 for (LL i=0;i<n;i++){
12     a[i] = S.lower_bound(a[i]) ->
13         second;
14 }
15 // find() return the iterator,
16 // then take the rank
17 // or S.find(a[i]) -> second;
18 }

```

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        k += k+1;
19        return k; //return the size of subset
20    }
21 }
22 LL exp_modp(LL x,LL y,LL p){
23     if(y == 0) return 1;
24     if(y % 2) return (exp_modp(x,y-1,p)*x
25         ) % p;
26     else{
27         LL temp = exp_modp(x,y/2,p);
28         return (temp*temp) % p;
29     }
30 }
31 int main(){
32     good;
33     //freopen("file name", "r", stdin);
34     //input redirection
35     LL i,n,p;
36     LL a[30],b[30];
37     cin >> n >> p;
38     int len_a = n/2,len_b = n - len_a;
39     for(i=0;i<len_a;i++){
40         cin >> a[i];
41     }
42     for(i=0;i<len_b;i++){
43         cin >> b[i];
44     }
45     LL len_sa = subset(a,len_a,sa,p);
46     LL len_sb = subset(b,len_b,sb,p);
47     sort(sa,sa+len_sa);
48     sort(sb,sb+len_sb);
49     LL len_sb2 = 1; //len_sb2 followed by
50     //i below
51     no[0] = 1; //assume not empty(check
52     //later)
53     for(i=1;i<len_sb;i++){
54         if(sb[i] != sb[i-1]){ //new
55             //element
56             sb[len_sb2] = sb[i];
57             no[len_sb2] = 1;
58             len_sb2++;
59         }
60         else //old element
61             no[len_sb2-1]++;
62     }
63     LL ans = (sb[0] == 1) ? no[0] % p : 0;
64     for(i=0;i<len_sa;i++){
65         if(sa[i] == 1) ans = (ans+1) % p;
66         LL y = exp_modp(sa[i],p-2,p); //
67         //module inverse
68         int it = lower_bound(sb,sb+
69             len_sb2,y) - sb;
70     }

```

```

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
11                  -1] + 1;
12           else
13              dp[i][j] = max(dp[i-1][j
14                  ], dp[i][j-1]);
15       }
16   }
17   cout << dp[text1.size()][text2.size()]
18   ;

```

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;
18 }

```

2.8 skew heap

```

1 node *merge(node *a, node *b){
2     if(!a || !b) return a ? 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 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] = 1;
10        sp.clear(); h.clear();
11    }
12    void assign(int *k, int v) {
13        h.pb({k, *k});
14        *k = v;
15    }
16    void save() { sp.pb(SZ(h)); }
17    void undo() {
18        assert(!sp.empty());
19        int last = sp.back(); sp.pop_back();
20        while (SZ(h) != last) {
21            auto x = h.back(); h.pop_back();

```

```

22        *x.F = x.S;
23    }
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         }
48         return false;
49     }
50     int main(){
51         good;
52         LL vertex, edge;
53         cin >> vertex >> edge;
54         memset(dis, maxdis, sizeof(dis));
55         vector<Node> graph(edge);
56         for(LL i = 0; i < edge; i++){
57             cin >> graph[i].in >> graph[i].
58             out >> graph[i].weight;
59         }
60         LL source;
61         cin >> source;
62         bellman_ford(graph, vertex, edge, source
63             );
64         if(checkinf(graph, edge))
65             cout << "found negative round" <<
66             endl;
67         else
68             cout << "not found negative round
69             " << endl;
70         return 0;
71     }

```

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 DAG 最長最短

```

1 #include <bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3   ; cin.tie(0)
4 #define sobig 1000000001
5 typedef long long LL;
6 using namespace std;
7
8 int main(){
9     good;
10    LL n, m, a, b, w, start, fin, cnt = 0;
11    cin >> n >> m;
12    cin >> start >> fin;
13    vector<pair<LL, LL>> adj[n]; // n-> out
14    //neighbor for first, weight for
15    //second
16    LL indeg[n] = {0};
17    LL shortest[n], longest[n]; //dis. from
18    //start to i
19    for(LL i = 0; i < m; i++){
20        cin >> a >> b >> w;
21        adj[a].push_back({b, w});
22        indeg[b]++;
23    }
24    queue<LL> Q; //manage tologpy
25    //sequence
26    for(LL i = 0; i < n; i++){
27        shortest[i] = sobig, longest[i] = -sobig;
28        shortest[start] = longest[start] = 0;
29        for(LL i = 0; i < n; i++){
30            if(indeg[i] == 0)
31                Q.push(i);
32        }
33        while(!Q.empty()){
34            LL v = Q.front();
35            Q.pop(), cnt++;
36            for(auto e: adj[v]){
37                if(shortest[v] < sobig){ //
38                    //path exist
39                    shortest[e.first] = min(
40                        shortest[e.first], e.
41                        second + shortest[v]
42                    );
43                    longest[e.first] = max(
44                        longest[e.first], e.
45                        second + longest[v]
46                    );
47                }
48                if(--indeg[e.first] == 0)
49                    Q.push(e.first);
50            }
51        }
52        if(cnt != n) cout << "not a DAG" <<
53        endl;
54        if(shortest[fin] == sobig)
55            cout << "No path" << endl << "No
56            path";
57        else
58            cout << shortest[fin] << endl <<
59            longest[fin];
60        return 0;
61    }

```

3.4 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 }
21 int main(){
22     good;
23     LL i,n,a,b;
24     cin >> n;
25     for(i = 0; i < n; i++){
26         cin >> a >> b;
27         v[a].push_back(b);
28         v[b].push_back(a);
29     }
30     dfs(0,1);
31     return 0;

```

3.5 dijkstra

```

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

```

3.6 topology sort

```

53 return 0;
54 }
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        if(tail < n) cout << "not a DAG" <<
26            endl;
27        else{
28            for(LL i = 0; i < n; i++){
29                cout << topo[i] << ' ';
30            }
31            return 0;
32        }
33    }

```

3.7 union and find

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3   ;cin.tie(0)
4 #define RSIZE 101
5 typedef long long LL;
6 using namespace std;
7 LL parent[503*503];
8 int graph[503*503] = {0};
9 int dxy[4] = {1,-1};
10 LL now_area = 0,max_area = 0;
11 LL sfind(LL dots){//find Leader,Leader's
12     parent = size of set
13     if(parent[dots] < 0)
14         return dots;
15     return parent[dots] = sfind(parent[
16         dots]);
17 }
18 LL BFS(LL now,LL root){//find root and
19     return size
20     parent[now] = root;
21     LL cnt = 1;
22     for(int k = 0; k < 4; k++){//4
23         directions
24         int u = now+dxy[k];
25         if(graph[u] == 1 && parent[u] ==
26             -1)//unvisited
27             cnt += BFS(u,root);
28     }
29     return cnt;
30 }
31 void combine(LL u,LL v){//merge two sets
32     LL set1 = sfind(u),set2 = sfind(v);
33     if(set1 == set2) return;//same set
34     max_area = max(max_area,-parent[set1]
35         ]-parent[set2]);
36     now_area--;//merge -> 2 pools become
37     1
38     if(parent[set1] < parent[set2]){//1
39         is larger
40         parent[set1] += parent[set2];
41         parent[set2] = set1;
42     }
43     else{
44         parent[set2] += parent[set1];
45         parent[set1] = set2;
46     }
47     return;
48 }

```

```

43 int main(){
44     good;
45     //freopen("file name", "r", stdin);
46     //input redirection
47     LL i,j,m,n,k;
48     cin >> m >> n >> k;
49     memset(parent,-1,sizeof(parent));
50     for(i = 1; i <= m; i++){
51         for(j = 1; j <= n; j++){
52             cin >> graph[i*(n+2)+j];
53         }
54     }
55     n += 2;
56     dxy[2] = n,dxy[3] = -n;
57     LL mn = (m+1)*n;
58     for(LL x = n; x < mn; x++){
59         if(graph[x] == 1 && parent[x] ==
60             -1){//unvisited
61             parent[x] = -BFS(x,x);//first
62             point consider as root
63             now_area++;
64             max_area = max(max_area,-
65                 parent[x]);
66         }
67     }
68     LL ans = now_area,max_ans = max_area;
69     while(k--){
70         LL x,y,temp;
71         cin >> x >> y;
72         temp = x*n+y;
73         if(graph[temp] == 1) continue;
74         graph[temp] = 1;
75         now_area++;
76         max_area = max(max_area,(LL)1);
77         for(i = 0; i < 4; i++){
78             if(graph[temp+dxy[i]] == 0)
79                 continue;
80             combine(temp,temp+dxy[i]);
81         }
82         ans += now_area;
83         max_ans += max_area;
84     }
85     cout << max_ans << endl << ans;
86     return 0;
87 }

```

3.8 棋格最少轉彎

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4   ;cin.tie(0)
5 typedef long long LL;
6 using namespace std;
7 int main(){
8     good;
9     //freopen("P_2_1_5.in", "r", stdin)
10    //input redirection
11    LL i,j,m,n;
12    LL dx[4] = {-1,0,1,0},dy[4] =
13        {0,1,0,-1};
14    cin >> m >> n;
15    char graph[m+5][n+5];
16    LL dis[m+2][n+2];
17    for(i = 0; i <= m+1; i++){
18        graph[i][0] = graph[i][n+1] = '1'
19        ;//Left/right bound
20    }
21    for(j = 0; j <= n+1; j++){
22        graph[0][j] = graph[m+1][j] = '1'
23        ;
24    }
25    for(i = 1; i <= m; i++){
26        for(j = 1; j <= n; j++){
27            cin >> graph[i][j];
28        }
29    }
30    for(i = 1; i <= m; i++){
31        for(j = 1; j <= n; j++){
32            dis[i][j] = -1;//not visited
33        }
34    }
35    queue<pair<LL,LL>> Q;//visit i,j
36    Q.push({1,1});
37    dis[1][1] = 0;
38    //start BFS
39    while(!Q.empty() && dis[m][n] < 0){
40        auto dots = Q.front();
41        Q.pop();
42        LL now_x = dots.first,now_y =
43            dots.second;
44        for(LL k = 0; k < 4; k++){
45            LL new_x = now_x+dx[k],new_y
46                = now_y+dy[k];
47            while (graph[new_x][new_y] ==
48                '0'){
49
50            }
51        }
52    }

```

```

39         if(dis[new_x][new_y] ==
40             -1){//not visited
41             dis[new_x][new_y] =
42                 dis[now_x][now_y
43                     ]+1;
44             Q.push({new_x,new_y})
45             ;
46         }
47     }
48     new_x += dx[k];
49     new_y += dy[k];
50 }
51 }

```

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 long long int phi[N+1];
8 void phiTable(){
9     for(int i=1; i<=N; i++) phi[i]=i;
10    for(int i=1; i<=N; i++) for(x=i*2; x<=N; x+=
11        i) phi[x]-=phi[i];
12 }
13 void all_divdown(const LL &n) {// all n/x
14     for(LL a=1; a<=n; a=n/(n/(a+1))) {
15         // dosomething;
16     }
17 }
18 const int MAXPRIME = 1000000;
19 int iscom[MAXPRIME], prime[MAXPRIME],
20     primecnt;
21 int phi[MAXPRIME], mu[MAXPRIME];
22 void sieve(void){
23     memset(iscom, 0, sizeof(iscom));
24     primecnt = 0;
25     phi[1] = mu[1] = 1;
26     for(int i=2; i<=MAXPRIME; ++i) {
27         if(!iscom[i]) {
28             prime[primecnt++] = i;
29             mu[i] = -1;
30             phi[i] = i-1;
31         }
32         for(int j=0; j<primecnt; ++j) {
33             int k = i * prime[j];
34             if(k>=MAXPRIME) break;
35             iscom[k] = prime[j];
36             if(i%prime[j]==0) {
37                 mu[k] = 0;
38                 phi[k] = phi[i] * prime[j];
39                 break;
40             } else {
41                 mu[k] = -mu[i];
42                 phi[k] = phi[i] * (prime[j]-1);
43             }
44         }
45     }
46 }
47 bool g_test(const LL &g, const LL &p,
48     const vector<LL> &v) {
49     for(int i=0; i<v.size(); ++i)
50         if(modexp(g, (p-1)/v[i], p)==1)
51             return false;
52     return true;
53 }
54 LL primitive_root(const LL &p) {
55     if(p==2) return 1;
56     vector<LL> v;
57     Factor(p-1, v);
58     v.erase(unique(v.begin(), v.end()), v.
59         end());
60     for(LL g=2; g<p; ++g)
61         if(g_test(g, p, v))
62             return g;
63     puts("primitive_root NOT FOUND");
64     return -1;
65 }
66 int Legendre(const LL &a, const LL &p) {
67     return modexp(a%p, (p-1)/2, p);
68 }

```

```

64 LL inv(const LL &a, const LL &n) {
65     LL d, x, y;
66     gcd(a, n, d, x, y);
67     return d==1 ? (x+n)%n : -1;
68 }
69 int inv[maxN];
70 LL invtable(int n, LL P){
71     inv[1]=1;
72     for(int i=2; i<n; ++i)
73         inv[i]=(P-(P/i))*inv[P%i]%P;
74 }
75 LL log_mod(const LL &a, const LL &b,
76     const LL &p) {
77     // a ^ x = b (mod p)
78     int m=sqrt(p+.5), e=1;
79     LL v=inv(modexp(a, m, p), p);
80     map<LL, int> x;
81     x[1]=0;
82     for(int i=1; i<m; ++i) {
83         e = LLMul(e, a, p);
84         if(!x.count(e)) x[e] = i;
85     }
86     for(int i=0; i<m; ++i) {
87         if(x.count(b)) return i*m + x[b];
88         b = LLMul(b, v, p);
89     }
90     return -1;
91 }
92 LL Tonelli_Shanks(const LL &n, const LL &
93     p) {
94     // x^2 = n (mod p)
95     if(n==0) return 0;
96     if(Legendre(n, p)!=1) while(1) { puts("
97         SQRRT ROOT does not exist"); }
98     int S = 0;
99     LL Q = p-1;
100    while( !(Q&1) ) { Q>>=1; ++S; }
101    if(S==1) return modexp(n%p, (p+1)/4, p);
102    LL z = 2;
103    for(; Legendre(z, p)!=-1; ++z)
104        LL c = modexp(z, Q, p);
105    LL R = modexp(n%p, (Q+1)/2, p), t =
106        modexp(n%p, Q, p);
107    int M = S;
108    while(1) {
109        if(t==1) return R;
110        LL b = modexp(c, 1<<(M-i-1), p);
111        R = LLMul(R, b, p);
112        t = LLMul(LLMul(b, b, p), t, p);
113        c = LLMul(b, b, p);
114        M = i;
115    }
116    return -1;
117 }
118 template<typename T>
119 T Euler(T n){
120     T ans=n;
121     for(T i=2; i*i<=n; ++i){
122         if(n%i==0){
123             ans=ans/(i-1);
124             while(n%i==0) n/=i;
125         }
126     }
127     if(n>1) ans=ans/n*(n-1);
128     return ans;
129 }
130 //Chinese_remainder_theorem
131 template<typename T>
132 T pow_mod(T n, T k, T m){
133     T ans=1;
134     for(n=(n>m?n%m:n); k>=1){
135         if(k&1) ans=ans*n%m;
136         n=n*n%m;
137         k>>=1;
138     }
139     return ans;
140 }
141 template<typename T>
142 T crt(vector<T> &m, vector<T> &a){
143     T M=1, tm, ans=0;
144     for(int i=0; i<(int)m.size(); ++i) M*=m[i];
145     for(int i=0; i<(int)a.size(); ++i){
146         tm=M/m[i];
147         ans=(ans+a[i]*tm%M)*pow_mod(tm, Euler
148             (m[i])-1, m[i])%M;
149     }
150     /*如果m[i]是質數 · Euler(m[i])-1=m[i]
151         j-2 · 就不用算Euler了*/
152     return ans;
153 }

```

```

153 //java code
154 //求sqrt(N)的連分數
155 public static void Pell(int n){
156     BigInteger N, p1, p2, q1, q2, a0, a1, a2, g1, g2
157         , h1, h2, p, q;
158     g1=q2=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         return ;
166     }
167     while(true){
168         g2=a1.multiply(h1).subtract(g1);
169         h2=N.subtract(g2.pow(2)).divide(h1);
170         a2=g2.add(a0).divide(h2);
171         p=a1.multiply(p2).add(p1);
172         q=a1.multiply(q2).add(q1);
173         if(p.pow(2).subtract(N.multiply(q.
174             pow(2))).compareTo(BigInteger.
175                 ONE)==0)break;
176         g1=g2; h1=h2; a1=a2;
177         p1=p2; p2=p;
178         q1=q2; q2=q;
179     }
180     System.out.println(p+" "+q);
181 }

```

4.2 bit set

```

1 void sub_set(int S){
2     int sub=S;
3     do{
4         //對某集合的子集的處理
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 << "gOHeRC2" << 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 }
18 LL pow(LL a, LL b, LL mod){ //a^b%mod
19     LL ans=1;
20     for(; b; a=a%mod, b>>=1)
21         if(b&1) ans=ans*a%mod;
22     return ans;
23 }
24 bool isless(LL *a, int n, int k){
25     if(*a==1) return k>1;
26     if(--n==0) return *a<k;
27     int next=0;
28     for(LL b=1; b<k; ++next)
29         b*=*a;
30     return isless(a+1, n, next);
31 }
32 LL high_pow(LL *a, int n, LL mod){
33     if(*a==1 || --n==0) return *a%mod;
34     int k=0, r=euler[mod];
35     for(LL tma=1; tma!=pow(*a, k+r, mod); ++k)
36         tma=tma*(*a)%mod;
37     if(isless(a+1, n, k)) return pow(*a,
38         high_pow(a+1, n, k), mod);
39     int tmd=high_pow(a+1, n, r), t=(tmd-k+r)%
40         r;
41     return pow(*a, k+t, mod);
42 }
43 LL a[1000005];
44 int t, mod;
45 int main(){
46     init_euler();
47     scanf("%d", &t);
48     #define n 4
49     while(t--){
50         for(int i=0; i<n; ++i) scanf("%lld", &a[i]);
51         scanf("%d", &mod);
52         printf("%lld\n", high_pow(a, n, mod));
53     }
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 }
10 LL calcmmod(LL index, LL p){
11     if(index == 0) return base[index] - '0';
12     LL single = calcmmod(index-1, p)*10;
13     return (single%p + base[index] - '0')%p;
14 }

```

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 }

```

```

8     if(y % 2) return (
9         mod_inverse_by_speed_exp(x, y-1, p)
10        ) * x % p;
11     else{
12         LL temp =
13             mod_inverse_by_speed_exp(x, y
14             /2, p);
15         return (temp*temp)%p;
16     }
17 }
18 int main(){
19     good;
20     LL n, i, p, x;
21     cin >> n >> p;
22     for(i = 0; i < n; i++){
23         cin >> x;
24         cout << mod_inverse_by_speed_exp(
25             x, p-2, p) << ' ';
26     }
27     return 0;
28 }

```

4.8 質因數分解

```

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 pollorrho(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(isp[(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 ) {
45     if(n<1e9) {
46         smallfactor(n, v);
47         return;
48     }
49     if(Isprime(n)) {
50         v.push_back(n);
51         return;
52     }
53     LL d;
54     for(int c=3; ; ++c) {
55         d = pollorrho(n, c);
56         if(d!=n) break;
57     }
58     comfactor(d, v);
59     comfactor(n/d, v);
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
70 void AllFactor(const LL &n, vector<LL> &v)
71 {
72     vector<LL> tmp;
73     Factor(n, tmp);
74     v.clear();
75     v.push_back(1);
76     int len;
77     LL now=1;
78     for(int i=0; i<tmp.size(); ++i) {
79         if(i==0 || tmp[i]!=tmp[i-1]) {
80             len = v.size();
81             now = 1;
82         }
83         now*=tmp[i];
84         for(int j=0; j<len; ++j)
85             v.push_back(v[j]*now);
86     }
87 }

```

5 String

5.1 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
10 struct {
11     LL center, height;
12     LL pre, next;
13     bool alive;
14 } tree[RSIZE];
15
16 queue<LL> Q; //check removed tree
17 void removable(LL index){
18     if(!tree[index].alive) return;
19     LL s = tree[index].pre, t = tree[index]
20         .next;
21     if(tree[index].center - tree[index].
22         height >= tree[s].center ||
23         tree[index].center + tree[index].
24         height <= tree[t].center){
25         tree[index].alive = false;
26         Q.push(index);
27         tree[s].next = t;
28         tree[t].pre = s;
29     }
30 }
31
32 int main(){
33     good;
34     LL bound, n, total = 0, high = 0;
35     cin >> n >> bound;
36     for(LL i = 1; i <= n; ++i)
37         cin >> tree[i].center;
38     for(LL i = 1; i <= n; ++i)
39         cin >> tree[i].height;
40     for(LL i = 1; i <= n; ++i){
41         tree[i].pre = i-1;
42         tree[i].next = i+1;
43         tree[i].alive = true;
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 }

```

5.2 manacher(最小回文字串)

```

1 //原字串: asdsasdsa
2 //要把字串變成這樣: @#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

```

```

48    sort(adjacent.begin(),adjacent.end(),
49          cmp); //sort by weight
50    LL cost = 0,now_edge = 0;
51    for(EDGE e : adjacent){
52        if(merge(e.u,e.v)){ //connect edge
53            cost += e.w;
54            now_edge++;
55        }
56        if(now_edge < n-1) //not a MST
57            cout << -1 << endl;
58        else
59            cout << cost << endl;
60        return 0;
61    }

```

```

34    if(visit[v]) continue;
35    visit[v] = true;
36    for(auto it : adjacent[v]){
37        LL neighbor = it.first,w = it.
38        second;
39        if(visit[neighbor]) continue;
40        if(w < dis[neighbor]){ //new
41            edge is shorter
42            dis[neighbor] = w;
43            fa[neighbor] = v;
44            PQ.push({-dis[neighbor],
45                neighbor});
46        }
47    }
48    LL cost = 0,cnt = 0;
49    //count cost and check if MST exists
50    for(i = 0; i < n; i++){
51        if(dis[i] < oo)
52            cost += dis[i];
53        else
54            cnt++;
55    }
56    if(cnt)
57        cout << -1 << endl;
58    else
59        cout << cost << endl;
60    return 0;

```

6 Tree Problem

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 }

```

6.3 LCA

```

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 {
8     pa[0][x]=p;
9     for(int i=0;i<=MLG;++i)
10        pa[i+1][x]=pa[i][pa[i][x]];
11    for(auto &i:G[x]){
12        if(i==p) continue;
13        dep[i]=dep[x]+1;
14        dfs(i,x);
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    }

```

6.5 segment tree

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 #define cin.tie(0)
4 #define RSIZE 100000
5 #define pll pair<LL,LL>
6 #define lc 2*index //c means child
7 #define rc 2*index+1
8 typedef long long LL;
9 using namespace std;
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[temp] = 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[temp] = 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
52 //ones are full range
53 LL query(LL l,LL r,LL L,LL R,LL index){
54     if(l <= L && R <= r) return tree[
55         index];
56     push(index); //if use single point
57     //modify,no need
58     LL M = (L+R)/2;

```

6.2 kruskal(MST)

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 #define 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
15 bool cmp(EDGE &a,EDGE &b){ //sort by
16     //weight
17     return a.w < b.w;
18 }
19 LL sfind(LL now){ //find root,root's
20     //father=set size
21     if(fa[now] < 0)
22         return now;
23     return fa[now] = sfind(fa[now]);
24 }
25 bool merge(LL u,LL v){ //find two root,
26     //comparing size(by root's father)
27     LL set1 = sfind(u),set2 = sfind(v);
28     if(set1 == set2) return false; //same
29     //root-> no need to merge
30     if(fa[set1] < fa[set2]){ //set1 is
31         //larger
32         fa[set1] += fa[set2];
33         fa[set2] = set1;
34     }
35     else{
36         fa[set2] += fa[set1];
37         fa[set1] = set2;
38     }
39     return true;
40 }
41 int main(){
42     good;
43     //freopen("file name", "r", stdin);
44     //input redirection
45     LL i,n,m;
46     cin >> n >> m;
47     for(i = 0; i < m; i++){
48         LL x,y,weight;
49         cin >> x >> y >> weight;
50         adjacent.push_back({x,y,weight});
51     }
52     memset(fa,-1,sizeof(fa)); //unvisited

```

6.4 Prim(MST)

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 #define 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
14 int main(){
15     good;
16     //freopen("file name", "r", stdin);
17     //input redirection
18     LL i,n,m;
19     cin >> n >> m;
20     for(i = 0; i < m; i++){
21         LL x,y,w;
22         cin >> x >> y >> w;
23         adjacent[x].push_back({y,w});
24         adjacent[y].push_back({x,w});
25     }
26     //initial
27     LL start = 0;
28     memset(dis,oo,sizeof(dis));
29     memset(fa,-1,sizeof(fa));
30     priority_queue<pair<LL,LL>> PQ; //dis
31     //[],vertex
32     PQ.push({dis[start] = 0,start});
33     //prim
34     while (!PQ.empty()){
35         auto pt = PQ.top();
36         PQ.pop();
37         LL v = pt.second;

```

```

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 int main(){
79 good;
80 LL k,n;
81 //build(1,n,1)at first,can use query(
l,r,1,n,1).
82 return 0;
83 }
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)
10 //same diagonal
11 valid = false;
12 break;
13 }
14 }
15 if(valid) total++;
16 } while (next_permutation(p,p+n));
17 return total;
18 }

```

7.2 debug

```

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

```

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 ;
6 -Wl,--stack,214748364 -trigraphs
7 #pragma comment(linker, "/STACK
    :1024000000,1024000000")
8 //linux stack resize
9 #include<sys/resource.h>
10 void increase_stack(){
11 const rlim_t ks=64*1024*1024;
12 struct rlimit rl;
13 int res=getrlimit(RLIMIT_STACK,&rl);
14 if(!res&&rl.rlim_cur<ks){
15 rl.rlim_cur=ks;
16 res=setrlimit(RLIMIT_STACK,&rl);
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 swepline

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4 ;cin.tie(0)
5 typedef long long LL;
6
7 using namespace std;
8
9 struct Seg{
10 LL left,right;
11 };
12 bool cmp(Seg &a,Seg &b){
13 return a.left < b.left;
14 }
15
16 int main(){
17 good;
18 //freopen("P_2_1_5.in", "r", stdin)
19 ;//input redirection
20 LL n;
21 cin >> n;
22 Seg line[n];
23 for(LL i = 0; i < n; i++){
24 cin >> line[i].left >> line[i].
    right;
25 sort(line,line+n,cmp);
26 Seg last = line[0];
27 LL total = 0;
28 for(LL i = 1; i < n; i++){
29 if(line[i].left > last.right){
30 total += last.right - last.
    left;
31 last = line[i];
32 continue;
33 }
34 last.right = max(last.right,line[
    i].right);//merge last and
line[i]
35 }
36 total += last.right - last.left;

```

```

36 cout << total;
37 return 0;
38 }

```

7.7 two number(another slide)

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4 ;cin.tie(0)
5 #define SIZE 100005
6 typedef long long LL;
7
8 using namespace std;
9
10 int main(){
11 good;
12 //freopen("P_2_1_5.in", "r", stdin)
13 ;//input redirection
14 LL i,j,m,n,k,ans = 0;
15 LL A[SIZE],B[SIZE];
16 cin >> m >> n >> k;
17 for(i = 0; i < m; i++)
18 cin >> A[i];
19 for(i = 0; i < n; i++)
20 cin >> B[i];
21 sort(A,A+m);
22 sort(B,B+n);
23 j = n-1;
24 for(i = 0; i < m; i++){
25 while(j > 0 && B[j] > k-A[i])
26 j--;
27 if(A[i] + B[j] == k)
28 ans++;
29 }
30 cout << ans << endl;
31 return 0;
32 }

```

7.8 模板

```

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 typedef long long LL;
9 using namespace std;
10
11 int main(){
12 good;
13 //freopen("file name", "r", stdin);
14 ;//input redirection
15 return 0;

```

8 other

8.1 1D0D dp

```

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
7 int main(){
8 good;
9 //freopen("file name", "r", stdin);
10 ;//input redirection
11 LL i,n;
12 LL money[100002],dp[100002] = {0};//dp
for last i day max profit
13 cin >> n;
14 for(i = 1; i <= n; i++)
15 cin >> money[i];
16 dp[0] = 0,dp[1] = money[1],dp[2] =
    max(money[1],money[2]);
17 for(i = 3; i <= n; i++)
18 dp[i] = max(dp[i-2]+money[i],dp[i
    -1]);//choose,discard
19 cout << dp[i-1];
20 return 0;

```

8.2 maximum subarray

```
1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4   ;cin.tie(0)
5
6 typedef long long LL;
7
8 using namespace std;
9
10 int main(){
11     good;
12     //freopen("P_2_1_5.in", "r", stdin)
13     ;//input redirection
14     LL n;
15     cin >> n;
16     LL num[n];
17     for(LL i = 0; i < n; i++){
18         cin >> num[i];
19         LL max_sum = 0, sum = 0; //allow 0
20         for(LL i = 0; i < n; i++){
21             sum = max(num[i], sum+num[i]); //
22             choose if restart
23             max_sum = max(sum, max_sum); //
24             update max subarray
25         }
26         cout << max_sum;
27         return 0;
28 }
```

8.3 mergesort

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define good ios_base::sync_with_stdio(0)
4   ;cin.tie(0)
5
6 #define NORMALSIZE 8
7 typedef long long LL;
8
9 void merge(LL *, LL, LL, LL); //prototype
10 void mergesort(LL *, LL, LL);
11
12 void mergesort(LL* arr, LL l, LL r){
13     LL m;
14     if(l < r){
15         m = (l+r)/2;
16         mergesort(arr, l, m); //sort left
17         subarray
18         mergesort(arr, m+1, r); //sort right
19         sunarray
20         merge(arr, l, m, r);
21     }
22     return;
23 }
24
25 void merge(LL* arr, LL l, LL m, LL r){
26     LL left, right, tmp[NORMALSIZE], i;
27     left = l; // left subarr.begin
28     right = m+1; //right subarr.begin
29     i = l;
30     while ((left <= m) && (right <= r)){
31         //merging left subarr and right
32         subarr
33         if(arr[left] < arr[right]){ //
34             left subarr is smaller
35             tmp[i] = arr[left];
36             i++, left++;
37         }
38         else{ //the other situation
39             tmp[i] = arr[right];
40             i++, right++;
41         }
42     }
43     while(left <= m){ // right subarr is
44         fully sorted
45         tmp[i] = arr[left];
46         i++, left++;
47     }
48     while(right <= r){ // left subarr is
49         fully sorted
50         tmp[i] = arr[right];
51         i++, right++;
52     }
53     for(i = l; i <= r; i++)
54         arr[i] = tmp[i];
55 }
```

8.4 sliding window

```
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 }
```

8.5 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
4         ==9&&d<3)
5         return (d+2*m+3*(m+1)/5+y+y/4+5)%7;
6     return (d+2*m+3*(m+1)/5+y+y/4-y/100+y
7         /400)%7;
8 }
```

8.6 逆序數對

```
1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4   ;cin.tie(0)
5
6 typedef long long LL;
7
8 using namespace std;
9 LL number[100002] = {0};
10
11 LL inverse_pair(LL a[], LL left, LL right){
12     LL i, j, mid = (left+right)/2;
13     if(left+1 >= right) return 0; //one
14     num has no pair
15     LL ans = inverse_pair(a, left, mid) +
16     inverse_pair(a, mid+1, right); //
17     pair in same side
18     LL cross = 0; //pair cross both side
19     sort(a+mid, a+right); //sort right part
20     of num
21     for(i = 0; i < mid; i++){
22         cross += lower_bound(a+mid, a+
23         right, a[i])-(a+mid); //
24         quantity of smaller than a[i]
25     }
26     return ans+cross;
27 }
28
29 int main(){
30     good;
31     LL n;
32     cin >> n;
33     for(LL i = 0; i < n; i++){
34         cin >> number[i];
35     }
36     cout << inverse_pair(number, 0, n);
37     return 0;
38 }
```

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 $Cv(G)$ ，最小邊覆蓋設為 $Ce(G)$ 。對於任意連通圖：

$$(a) I(G) + Cv(G) = |V|$$

$$(b) M(G) + Ce(G) = |V|$$

- 對於連通二分圖：

$$(a) I(G) = Cv(G)$$

$$(b) M(G) = Ce(G)$$

- 最大權閉合圖：

$$(a) C(u, v) = \infty, (u, v) \in E$$

$$(b) C(S, v) = W_v, W_v > 0$$

$$(c) C(v, T) = -W_v, W_v < 0$$

$$(d) ans = \sum_{W_v > 0} W_v - flow(S, T)$$

- 最大密度子圖：

$$(a) \text{求 } \max \left(\frac{W_e + W_v}{|V|} \right), e \in E', v \in V'$$

$$(b) U = \sum_{v \in V} 2W_v + \sum_{e \in E} W_e$$

$$(c) C(u, v) = W_{(u, v)}, (u, v) \in E \cdot \text{雙向邊}$$

$$(d) C(S, v) = U, v \in V$$

$$(e) D_u = \sum_{(u, v) \in E} W_{(u, v)}$$

$$(f) C(v, T) = U + 2g - D_v - 2W_v, v \in V$$

$$(g) \text{二分搜 } g:$$

$$l = 0, r = U, eps = 1/n^2$$

$$\text{if}((U \times |V| - flow(S, T))/2 > 0) \text{ } l =$$

$$mid$$

$$\text{else } r = mid$$

$$(h) ans = min_cut(S, T)$$

$$(i) |E| = 0 \text{ 要特殊判斷}$$

- 弦圖：

$$(a) \text{點數大於 } 3 \text{ 的環都要有一條弦}$$

$$(b) \text{完美消除序列從後往前依次給每個點染色：給每個點染上可以染的最小顏色}$$

$$(c) \text{最大團大小} = \text{色數}$$

$$(d) \text{最大獨立集：完美消除序列從前往後能選就選}$$

$$(e) \text{最小團覆蓋：最大獨立集的點和他延伸的邊構成}$$

$$(f) \text{區間圖是弦圖}$$

$$(g) \text{區間圖的完美消除序列：將區間按造又端點由小到大排序}$$

$$(h) \text{區間圖染色：用線段樹做}$$

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

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

- $D(i, g) = B_i - g \times C_i$
- $f(g) = \sum D(i, g) x_i$
- $f(g) = 0$ 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 $M(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$

9.1.9 Burnside's lemma

- $|X/G| = \frac{1}{|G|} \sum_{g \in G} |X^g|$
- $X^g = t^{c(g)}$
- 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。
- 正立方體塗 3 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 $\frac{1}{57} (3^6 + 6 \times 3^3 + 3 \times 3^4 + 8 \times 3^2 + 6 \times 3^3) =$

9.1.6 基本數論

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

9.1.10 Count on a tree

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

9.1.7 排組公式

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

9.1.8 冪次, 冪次和

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

Codebook - ss

Contents

1 Computational Geometry	1	3.6 topology sort	3	7.3 IncStack	7
1.1 SmallestCircle	1	3.7 union and find	3	7.4 input	7
1.2 最近點對	1	3.8 棋格最少轉彎	3	7.5 randomize	7
2 Data Structure	1	4 Number Theory	4	7.6 sweepline	7
2.1 01 背包	1	4.1 basic	4	7.7 two number(another slide)	7
2.2 array simulate linked list .	1	4.2 bit set	4	7.8 模板	7
2.3 binary search	1	4.3 matrix exponential	4	8 other	7
2.4 discretization	1	4.4 SpeedExpo	4	8.1 1D0D dp	7
2.5 half enumeration	1	4.5 外星模運算	5	8.2 maximum subarray	8
2.6 LCS	2	4.6 大數取模	5	8.3 mergesort	8
2.7 LIS	2	4.7 模逆元	5	8.4 sliding window	8
2.8 skew heap	2	4.8 質因數分解	5	8.5 WhatDay	8
2.9 undo disjoint set	2	5 String	5	8.6 逆序數對	8
3 Graph	2	5.1 linked list	5	9 zformula	8
3.1 bellman ford	2	5.2 manacher(最小回文字串)	6	9.1 formula	8
3.2 BFS	2	6 Tree Problem	6	9.1.1 Pick 公式	8
3.3 DAG 最長最短	2	6.1 findLCA	6	9.1.2 圖論	8
3.4 DFS	3	6.2 kruskal(MST)	6	9.1.3 dinic 特殊圖複雜度	8
3.5 dijkstra	3	6.3 LCA	6	9.1.4 0-1 分數規劃	8
		6.4 Prim(MST)	6	9.1.5 學長公式	9
		6.5 segment tree	6	9.1.6 基本數論	9
		7 default	7	9.1.7 排組公式	9
		7.1 8 queen	7	9.1.8 冪次, 冪次和	9
		7.2 debug	7	9.1.9 Burnside's lemma .	9
				9.1.10 Count on a tree . .	9

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 }
```