

1 Data Structure

1.1 01 背包

```

1 LL dp[101][100001] = {0}; //前i個物品所湊
  出重量j的最大價值
2 int main(){
3     good;
4     LL j,i,n,w,svalue = 0,sweight = 0;
5     cin >> n >> w;
6     pair<LL,LL> item[n+1]; //weight,value;
7     for(i = 1; i <= n; i++){
8         cin >> item[i].first;
9         for(i = 1; i <= n; i++){
10            cin >> item[i].second;
11        }
12        for(i = 0; i <= n; i++){
13            dp[i][0] = dp[0][i] = 0;
14            for(i = 1; i <= n; i++){
15                for(j = 1; j <= w; j++){
16                    if(item[i].first > j)
17                        dp[i][j] = dp[i-1][j];
18                    else
19                        dp[i][j] = max(dp[i-1][j],
20                                     item[i].second +
21                                     dp[i-1][j-item[i].
22                                     first]);
23                }
24            }
25        }
26        cout << dp[n][w];
27        return 0;
28    }
29 }

```

1.2 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 #define 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
15 queue<LL> Q; //check removed tree
16 void removable(LL index){
17     if(!tree[index].alive) return;
18     LL s = tree[index].pre,t = tree[index].next;
19     if(tree[index].center-tree[index].height >= tree[s].center ||
20        tree[index].center+tree[index].height <= tree[t].center){
21         tree[index].alive = false;
22         Q.push(index);
23         tree[s].next = t;
24         tree[t].pre = s;
25     }
26 }
27
28 int main(){
29     good;
30     LL bound,n,total = 0,high = 0;
31     cin >> n >> bound;
32     for(LL i = 1; i <= n; i++){
33         cin >> tree[i].center;
34         for(LL i = 1; i <= n; i++){
35             cin >> tree[i].height;
36             for(LL i = 1; i <= n; i++){
37                 tree[i].pre = i-1;
38                 tree[i].next = i+1;
39                 tree[i].alive = true;
40             }
41         }
42         tree[0].center = 0,tree[0].height = oo;
43         tree[n+1].center = bound,tree[n+1].height = oo;
44         for(LL i = 1; i <= n; i++){
45             removable(i);
46             while(!Q.empty()){
47                 LL v = Q.front();
48                 Q.pop();
49                 total++;
50                 high = max(high,tree[v].height);
51                 removable(tree[v].pre);

```

```

35         removable(tree[v].next);
36     }
37     cout << total << endl << high;
38     return 0;
39 }

```

1.3 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 }

```

1.4 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        // find() return the iterator,
14        then take the rank
15        // or S.find(a[i]) -> second;
16    }

```

1.5 half enumeration

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4 #define cin.tie(0)
5 typedef long long LL;
6 using namespace std;
7
8 LL sa[1<<18],sb[1<<18],no[1<<18]; //subset
9 product of a and b
10
11 LL subset(LL num[],LL length,LL product
12 [],LL p){ //pass by pointer
13     LL k = 0,i,j; //count
14     for(i = 0; i < length; i++){
15         for(j = 0; j < k; j++){
16             product[k+j] = (product[j]*
17                             num[i]) % p; //old
18             product times num[i]
19         }
20         product[k] = num[i]; //for num[i]
21         k += k+1;
22         return k; //return the size of subset
23     }
24 }
25
26 LL exp_modp(LL x,LL y,LL p){
27     if(y == 0) return 1;
28     if(y % 2) return (exp_modp(x,y-1,p)*x
29 ) % p;
30     else{
31         LL temp = exp_modp(x,y/2,p);
32         return (temp*temp) % p;
33     }
34 }
35
36 int main(){
37     good;
38     LL i,n,p;
39     LL a[30],b[30];
40     cin >> n >> p;
41     int len_a = n/2,len_b = n - len_a;
42     for(i = 0; i < len_a; i++)

```

```

36     cin >> a[i];
37     for(i = 0; i < len_b; i++)
38         cin >> b[i];
39     LL len_sa = subset(a,len_a,sa,p);
40     LL len_sb = subset(b,len_b,sb,p);
41     sort(sa,sa+len_sa);
42     sort(sb,sb+len_sb);
43
44     LL len_sb2 = 1; //len_sb2 followed by
45     i below
46     no[0] = 1; //assume not empty(check
47     later)
48     for(i = 1; i < len_sb; i++){
49         if(sa[i] != sb[i-1]){//new
50             element
51             sb[len_sb2] = sb[i];
52             no[len_sb2] = 1;
53             len_sb2++;
54         }
55         else{//old element
56             no[len_sb2-1]++;
57         }
58     }
59     LL ans = (sb[0] == 1) ? no[0] % p : 0;
60     for(i = 0; i < len_sa; i++){
61         if(sa[i] == 1) ans = (ans+1) % p;
62         LL y = exp_modp(sa[i],p-2,p); //
63         module inverse
64         int it = lower_bound(sa,sa+
65         len_sb2,y) - sa;
66         if(it < len_sb2 && sb[it] == y){
67             ans = (ans + no[it]) % p;
68         }
69     }
70     cout << ans << '\n';
71     return 0;
72 }

```

1.6 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     ];

```

1.7 LIS

```

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

```

1.8 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 Graph

2.1 bellman ford

```

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 #define maxdis 1000000001
8 typedef long long LL;
9 using namespace std;
10 struct Node{
11     LL in,out,weight;
12 };
13 LL dis[RSIZE];
14 void bellman_ford(vector<Node> &edges,LL
   v,LL e,LL start){//v for vertex, e
   for edge
15 //vector<LL> dis(v,LLONG_MAX);
16 dis[start] = 0;
17 //relaxation
18 for(LL i = 0; i < v-1; i++){
19     for(LL j = 0; j < e; j++){
20         LL x = edges[j].in,y = edges[
           j].out;
21         LL w = edges[j].weight;
22         if(dis[x] != maxdis && dis[x]
           +w < dis[y])
23             dis[y] = dis[x] + w;
24     }
25 }
26 //check -weight round
27 cout << "from " << start << "'s
   shortest path to each vertexes
   is:" << endl;
28 for(LL i = 0; i < v; i++){
29     cout << i << "\t\t" << dis[i] <<
       endl;
30 }
31 }
32 bool checkinf(vector<Node> &edges,LL e){
33     for(LL i = 0; i < e; i++){
34         LL x = edges[i].in,y = edges[i].
           out;
35         LL w = edges[i].weight;
36         if(dis[x] != maxdis && dis[x] + w
           < dis[y])
37             return true;
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].
           out >> graph[i].weight;
49     }
50     LL source;
51     cin >> source;
52     bellman_ford(graph,vertex,edge,source
       );
53     if(checkinf(graph,edge))
54         cout << "found negative round" <<
           endl;
55     else
56         cout << "not found negative round
           " << endl;
57     return 0;
}

```

2.2 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 }

```

2.3 DAG 最長最短

```

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

```

2.4 DFS

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(0);cout.tie(0)
3 typedef long long LL;
4 using namespace std;
5 int fa[100000],d[100000] = {0};//
       unnecessary
6 bool visit[100000] = {false};
7 vector<LL> v[100000];
8 void dfs(LL now,LL depth){
9     for(auto x:v[now]){
10        if(!visit[x]){
11            cout << x << ' ';
12            visit[x] = true;
13            d[x] = depth;
14            fa[x] = now;
15            dfs(x,depth+1);

```

```

16        }
17    }
18 }
19 int main(){
20     good;
21     LL i,n,a,b;
22     cin >> n;
23     for(i = 0; i < n; i++){
24         cin >> a >> b;
25         v[a].push_back(b);
26         v[b].push_back(a);
27     }
28     dfs(0,1);
29     return 0;
30 }

```

2.5 dijkstra

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(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
       neighbor,weight of edge
9 LL dis[N],parent[N];
10 bool visit[N] = {false};
11
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
       [,vertex · 技巧性讓最小值pop
26     PQ.push({dis[source] = 0,source});
27     //dijkstra
28     while (!PQ.empty()){
29         auto p = PQ.top();
30         PQ.pop();
31         LL v = p.second;//vertex
32         if(visit[v]) continue;
33         visit[v] = true;
34         for(auto it : adjacent[v]){
35             LL e = it.first,w = it.second
           ;
36             if(w + dis[v] < dis[e]){
37                 dis[e] = w + dis[v];
38                 parent[e] = v;
39                 PQ.push({-dis[e],e});
40             }
41         }
42     }
43     LL maxd = -1,cnt = 0,far;
44     for(i = 0; i < n; i++){
45         if(dis[i] < oo){
46             if(dis[i] > maxd)
47                 maxd = dis[i],far = i;
48         }
49         else
50             cnt++;//for can't reach
51     }
52     cout << maxd << endl << cnt;
53     return 0;
54 }

```

2.6 floyd warshall

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
   ;cin.tie(0)
3 #define RSIZE 502
4 #define pll pair<LL,LL>
5 #define lc 2*index
6 #define rc 2*index+1
7 #define far 1000000001//1e9+1
8 typedef long long LL;
9 using namespace std;
10

```

```

11 LL dis[RSIZE][RSIZE];
12 vector<pll> adjacent[RSIZE];//out
13   neighbor,weight
14 void floyd_warshall(LL vertex){
15     LL t,k,i,j;
16     for(t = 1; t <= vertex; t++){
17         dis[t][t] = 0;
18         for(k = 1; k <= vertex; k++){
19             for(i = 1; i <= vertex; i++){
20                 for(j = 1; j <= vertex; j++){
21                     if(dis[i][k] + dis[k][j]
22                        < dis[i][j])
23                         dis[i][j] = dis[i][k]
24                           +dis[k][j];
25                 }
26             }
27         }
28     }
29     return;
30 }
31 int main(){
32     good;
33     LL m,n,x;
34     cin >> n >> m >> x;//n nodes, m edges
35     ,from 1 to n
36     memset(dis,0,sizeof(dis));
37     for(LL i = 0; i < m; i++){
38         LL a,b,w;
39         cin >> a >> b >> w;
40         adjacent[a].push_back({b,w});
41         dis[a][b] = w;
42     }
43     return 0;
44 }

```

2.7 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    return 0;
33 }

```

2.8 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 }

```

```

17 LL BFS(LL now,LL root){//find root and
18     return size
19     parent[now] = root;
20     LL cnt = 1;
21     for(int k = 0; k < 4; k++){///4
22         directions
23         int u = now+dxy[k];
24         if(graph[u] == 1 && parent[u] ==
25            -1)//unvisited
26             cnt += BFS(u,root);
27     }
28     return cnt;
29 }
30 void combine(LL u,LL v){//merge two sets
31     LL set1 = sfind(u),set2 = sfind(v);
32     if(set1 == set2) return;//same set
33     max_area = max(max_area,-parent[set1]
34                    ]-parent[set2]);
35     now_area--;//merge -> 2 pools become
36     1
37     if(parent[set1] < parent[set2]){///1
38         is larger
39         parent[set1] += parent[set2];
40         parent[set2] = set1;
41     }
42     else{
43         parent[set2] += parent[set1];
44         parent[set1] = set2;
45     }
46     return;
47 }
48 int main(){
49     good;
50     //freopen("file name", "r", stdin);
51     //input redirection
52     LL i,j,m,n,k;
53     cin >> m >> n >> k;
54     memset(parent,-1,sizeof(parent));
55     for(i = 1; i <= m; i++){
56         for(j = 1; j <= n; j++){
57             cin >> graph[i*(n+2)+j];
58         }
59     }
60     n += 2;
61     dxy[2] = n,dxy[3] = -n;
62     LL mn = (m+1)*n;
63     for(LL x = n; x < mn; x++){
64         if(graph[x] == 1 && parent[x] ==
65            -1)//unvisited
66             parent[x] = -BFS(x,x);//first
67             point consider as root
68             now_area++;
69             max_area = max(max_area,-
70                parent[x]);
71     }
72     LL ans = now_area,max_ans = max_area;
73     while(k--){
74         LL x,y,temp;
75         cin >> x >> y;
76         temp = x*n+y;
77         if(graph[temp] == 1) continue;
78         graph[temp] = 1;
79         now_area++;
80         max_area = max(max_area,(LL)1);
81         for(i = 0; i < 4; i++){
82             if(graph[temp+dxy[i]] == 0)
83                 continue;
84             combine(temp,temp+dxy[i]);
85         }
86         ans += now_area;
87         max_ans += max_area;
88     }
89     cout << max_ans << endl << ans;
90     return 0;
91 }

```

2.9 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
8 int main(){
9     good;
10    LL i,j,m,n;
11    LL dx[4] = {-1,0,1,0},dy[4] =
12        {0,1,0,-1};
13    cin >> m >> n;
14    char graph[m+5][n+5];

```

```

14 LL dis[m+2][n+2];
15 for(i = 0; i <= m+1; i++)
16     graph[i][0] = graph[i][n+1] = '1'
17     ;//Left/right bound
18 for(j = 0; j <= n+1; j++)
19     graph[0][j] = graph[m+1][j] = '1'
20     ;
21 for(i = 1; i <= m; i++)
22     for(j = 1; j <= n; j++){
23         cin >> graph[i][j];
24     }
25 for(i = 1; i <= m; i++){
26     for(j = 1; j <= n; j++){
27         dis[i][j] = -1;//not visited
28     }
29 }
30 queue<pair<LL,LL>> Q;//visit i,j
31 Q.push({1,1});
32 dis[1][1] = 0;
33 //start BFS
34 while(!Q.empty() && dis[m][n] < 0){
35     auto dots = Q.front();
36     Q.pop();
37     LL now_x = dots.first,now_y =
38         dots.second;
39     for(LL k = 0; k < 4; k++){
40         LL new_x = now_x+dx[k],new_y
41             = now_y+dy[k];
42         while (graph[new_x][new_y] ==
43            '0'){
44             if(dis[new_x][new_y] ==
45                -1)//not visited
46                 dis[new_x][new_y] =
47                     dis[now_x][now_y]
48                     +1;
49                 Q.push({new_x,new_y});
50             }
51         }
52         new_x += dx[k];
53         new_y += dy[k];
54     }
55 }
56 if(dis[m][n] > 0) dis[m][n]--;//count
57 one more
58 cout << dis[m][n];
59 return 0;
60 }

```

3 Number Theory

3.1 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=comb<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 }

```

3.2 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;

```

```

15 cin >> n;
16 cout << "90HRC²" << n+2 << "9μ-°";
17 exp(u,n);
18 }

```

3.3 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 }

```

3.4 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 }

```

3.5 模逆元

```

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

```

4 String

4.1 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 #define 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
15 queue<LL> Q;//check removed tree
16 void removable(LL index){
17     if(!tree[index].alive) return;
18     LL s = tree[index].pre,t = tree[index].next;
19     if(tree[index].center-tree[index].height >= tree[s].center || tree[index].center+tree[index].height <= tree[t].center){
20         tree[index].alive = false;
21         Q.push(index);
22         tree[s].next = t;
23         tree[t].pre = s;
24     }
25 }
26
27 int main(){
28     good;
29     LL bound,n,total = 0,high = 0;
30     cin >> n >> bound;
31     for(LL i = 1; i <= n; i++){
32         cin >> tree[i].center;
33         for(LL i = 1; i <= n; i++){
34             cin >> tree[i].height;
35             for(LL i = 1; i <= n; i++){
36                 tree[i].pre = i-1;
37                 tree[i].next = i+1;
38                 tree[i].alive = true;
39             }
40         }
41         tree[0].center = 0,tree[0].height = oo;
42         tree[n+1].center = bound,tree[n+1].height = oo;
43         for(LL i = 1; i <= n; i++){
44             removable(i);
45             while (!Q.empty()){
46                 LL v = Q.front();
47                 Q.pop();
48                 total++;
49                 high = max(high,tree[v].height);
50                 removable(tree[v].pre);
51                 removable(tree[v].next);
52             }
53             cout << total << endl << high;
54             return 0;
55 }

```

4.2 manacher(最小回文字串)

```

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 }

```

5 Tree Problem

5.1 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 }

```

5.2 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 #define typedef long long LL;
7 using namespace std;
8
9 struct EDGE{
10     LL u,v,w;
11 };
12 vector<EDGE> adjacent;//out neighbor, weight
13 LL fa[RSIZE];
14
15 bool cmp(EDGE &a,EDGE &b){//sort by weight
16     return a.w < b.w;
17 }
18 LL sfind(LL now){//find root,root's father=set size
19     if(fa[now] < 0)
20         return now;
21     return fa[now] = sfind(fa[now]);
22 }
23 bool merge(LL u,LL v){//find two root, comparing size(by root's father)
24     LL set1 = sfind(u),set2 = sfind(v);
25     if(set1 == set2) return false;//same root-> no need to merge
26     if(fa[set1] < fa[set2]){ //set1 is larger
27         fa[set1] += fa[set2];
28         fa[set2] = set1;
29     }
30     else{
31         fa[set2] += fa[set1];
32         fa[set1] = set2;
33     }
34     return true;
35 }
36
37 int main(){
38     good;
39     LL i,n,m;
40     cin >> n >> m;
41     for(i = 0; i < m; i++){
42         LL x,y,weight;
43         cin >> x >> y >> weight;
44         adjacent.push_back({x,y,weight});
45     }
46     memset(fa,-1,sizeof(fa));//unvisited
47     sort(adjacent.begin(),adjacent.end(),cmp);//sort by weight
48     LL cost = 0,now_edge = 0;
49     for(EDGE e : adjacent){
50         if(merge(e.u,e.v)){//connect edge
51             cost += e.w;
52             now_edge++;
53         }
54     }
55     if(now_edge < n-1){//not a MST
56         cout << -1 << endl;
57     }
58     else{
59         cout << cost << endl;
60         return 0;
61 }

```

5.3 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     pa[0][x]=p;
8     for(int i=0;i<MLG;i++){
9         pa[i+1][x]=pa[i][pa[i][x]];
10    }
11    for(auto &i:G[x]){
12        if(i==p)continue;
13        dep[i]=dep[x]+1;
14        dfs(i,x);
15    }
16 }
17 inline int jump(int x,int d){
18     for(int i=0;i<MLG;i++){
19         if((d>>i)&1) x=pa[i][x];
20         return x;
21     }
22 }
23 inline int find_lca(int a,int b){
24     if(dep[a]>dep[b])swap(a,b);
25     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 }

```

5.4 Prim(MST)

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 #define RSIZE 502
4 #define oo 1000000001 //1e9+1
5 typedef long long LL;
6 using namespace std;
7
8 vector<pair<LL,LL>> adjacent[RSIZE]; //out
9 LL dis[RSIZE],fa[RSIZE]; //dis for weight
10 bool visit[RSIZE] = {false};
11
12 int main(){
13     good;
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 start = 0;
24     memset(dis,oo,sizeof(dis));
25     memset(fa,-1,sizeof(fa));
26     priority_queue<pair<LL,LL>> PQ; //dis
27     PQ.push({dis[start] = 0,start});
28     //prim
29     while (!PQ.empty()){
30         auto pt = PQ.top();
31         PQ.pop();
32         LL v = pt.second;
33         if(visit[v]) continue;
34         visit[v] = true;
35         for(auto it : adjacent[v]){
36             LL neibor = it.first,w = it.second;
37             if(visit[neibor]) continue;
38             if(w < dis[neibor]){ //new
39                 //edge is shorter
40                 dis[neibor] = w;
41                 fa[neibor] = v;
42                 PQ.push({-dis[neibor],neibor});
43             }
44         }
45         LL cost = 0,cnt = 0;
46         //count cost and check if MST exists
47         for(i = 0; i < n; i++){
48             if(dis[i] < oo)
49                 cost += dis[i];
50             else
51                 cnt++;
52         }
53         if(cnt)
54             cout << -1 << endl;
55         else
56             cout << cost << endl;
57         return 0;
58 }

```

5.5 segment tree

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(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 {
30     if(L == R){
31         tree[index] = v;
32         return;
33     }
34     LL M = (L+R)/2;
35     if(x <= M) //left side
36         modify(x,v,L,M,lc);
37     else
38         modify(x,v,M+1,R,rc);
39     tree[index] = max(tree[lc],tree[rc]);
40 }
41 //a range including index has to add tag
42 void addtag(LL tag,LL index){
43     tree[index] += tag;
44     lazy[index] += tag;
45 }
46 //transferring tag to child
47 void push(LL index){
48     addtag(lazy[index],lc);
49     addtag(lazy[index],rc);
50     lazy[index] = 0; //tag is transfered
51     to child
52 }
53 //Lower variables are queried range,UPPER
54 //ones are full range
55 LL query(LL l,LL r,LL L,LL R,LL index){
56     if(l <= L && R <= r) return tree[
57         index];
58     push(index); //if use single point
59     //modify,no need
60     LL M = (L+R)/2;
61     if(r <= M) //answer in the left side,
62         //don't need to query right side
63         return query(l,r,L,M,lc);
64     else if(l > M) //in right side
65         return query(l,r,M+1,R,rc);
66     else //answer cross both side
67         return max(query(l,r,L,M,lc),
68             query(l,r,M+1,R,rc)); //
69         choose better one
70 }
71 void multi_modify(LL l,LL r,LL v,LL L,LL
72 R,LL index){
73     if(l <= L && R <= r){
74         addtag(v,index);
75         return;
76     }
77     push(index);
78     LL M = (L+R)/2;
79     if(r <= M) multi_modify(l,r,v,L,M,lc)
80     ;
81     else if(l > M) multi_modify(l,r,v,M
82 +1,R,rc);
83     else{
84         multi_modify(l,r,v,L,M,lc);
85         multi_modify(l,r,v,M+1,R,rc);
86     }
87     tree[index] = max(tree[lc],tree[rc]);
88 }
89
90 int main(){
91     good;
92     LL k,n;
93     //build(1,n,1) at first,can use query(
94     //l,r,1,n,1).
95     return 0;
96 }

```

6 default

6.1 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 }
6 do{
7     bool valid = true;
8     for(int i = 0; i < n; i++){
9         for(int j = i+1; j < n; j++){
10             if(abs(p[i]-p[j]) == j-i)
11                 //same diagonal
12                 valid = false;
13                 break;
14             }
15         }
16         if(valid) total++;
17     } while (next_permutation(p,p+n));
18 return total;

```

6.2 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++;

```

6.3 sweepline

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(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 struct Seg{
11     LL left,right;
12 };
13 bool cmp(Seg &a,Seg &b){
14     return a.left < b.left;
15 }
16
17 int main(){
18     good;
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].
24         right;
25     }
26     sort(line,line+n,cmp);
27     Seg last = line[0];
28     LL total = 0;
29     for(LL i = 1; i < n; i++){
30         if(line[i].left > last.right){
31             total += last.right - last.
32             left;
33             last = line[i];
34             continue;
35         }
36         last.right = max(last.right,line[
37         i].right); //merge last and
38         line[i]
39     }
40     total += last.right - last.left;
41     cout << total;
42     return 0;

```

6.4 two number(another slide)

```

1 #include<bits/stdc++.h>
2
3 #define good ios_base::sync_with_stdio(0)
4 #define SIZE 100005
5 typedef long long LL;
6 using namespace std;
7
8 int main(){
9     good;
10     LL i,j,m,n,k,ans = 0;
11     LL A[SIZE],B[SIZE];
12 }

```



```

13 cin >> m >> n >> k;
14 for(i = 0; i < m; i++)
15     cin >> A[i];
16 for(i = 0; i < n; i++)
17     cin >> B[i];
18 sort(A,A+m);
19 sort(B,B+n);
20 j = n-1;
21 for(i = 0; i < m; i++){
22     while(j > 0 && B[j] > k-A[i])
23         j--;
24     if(A[i] + B[j] == k)
25         ans++;
26 }
27 cout << ans << endl;
28 return 0;
29 }

```

```

15 return memo[times] = sum;
16 }
17 int main(){
18     good;
19     //freopen("file name", "r", stdin);
20     //input redirection
21     LL n;
22     for(LL i = 1; i <= RSIZE; i++)
23         memo[i] = -1;
24     memo[0] = 1;
25     cin >> n;
26     cout << cata(n);
27     return 0;
28 }

```

```

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
15         mergesort(arr,m+1,r); //sort right
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
27         //subarr
28         if(arr[left] < arr[right]){ //
29             //left subarr is smaller
30             tmp[i] = arr[left];
31             i++;left++;
32         }
33         else{ //the other situation
34             tmp[i] = arr[right];
35             i++;right++;
36         }
37     }
38     while(left <= m){ // right subarr is
39         //fully sorted
40         tmp[i] = arr[left];
41         i++;left++;
42     }
43     while(right <= r){ // left subarr is
44         //fully sorted
45         tmp[i] = arr[right];
46         i++;right++;
47     }
48     for(i = l; i <= r; i++)
49         arr[i] = tmp[i];
50 }

```

6.5 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 oo 1000000001//1e9+1
6 #define pll pair<LL,LL>
7 #define lc 2*index
8 #define rc 2*index+1
9 typedef long long LL;
10 using namespace std;
11
12 int main(){
13     good;
14     return 0;
15 }

```

7.3 2D0D dp

```

1 #include<bits/stdc++.h>
2 #define good ios_base::sync_with_stdio(0)
3 ;cin.tie(0)
4 #define RSIZE 202
5 typedef long long LL;
6 using namespace std;
7
8 int main(){
9     good;
10    //freopen("file name", "r", stdin);
11    //input redirection
12    LL graph[RSIZE][RSIZE] = {0};
13    LL m,n,i,j;
14    cin >> m >> n;
15    for(i = 0; i < m; i++){ //0 base draw
16        graph
17        for(j = 0; j < n; j++)
18            cin >> graph[i][j];
19    }
20    for(i = 1; i < m; i++)
21        for(j = 1; j < n; j++)
22            graph[i][j] += graph[i-1][j];
23    for(j = 1; j < n; j++)
24        for(i = 1; i < m; i++)
25            graph[i][j] += graph[i][j-1];
26    for(i = 1; i < m; i++)
27        for(j = 1; j < n; j++)
28            graph[i][j] += graph[i-1][j-1];
29    cout << graph[m-1][n-1];
30    return 0;
31 }

```

7 other

7.1 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
13    //for last i day max profit
14    cin >> n;
15    for(i = 1; i <= n; i++)
16        cin >> money[i];
17    dp[0] = 0,dp[1] = money[1],dp[2] =
18    max(money[1],money[2]);
19    for(i = 3; i <= n; i++)
20        dp[i] = max(dp[i-2]+money[i],dp[i-1]);
21    //choose,discard
22    cout << dp[i-1];
23    return 0;
24 }

```

7.4 maximum subarray

```

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
8 int main(){
9     good;
10    //freopen("P_2_1_5.in", "r", stdin)
11    //input redirection
12    LL n;
13    cin >> n;
14    LL num[n];
15    for(LL i = 0; i < n; i++)
16        cin >> num[i];
17    LL max_sum = 0,sum = 0; //allow 0
18    for(LL i = 0; i < n; i++){
19        sum = max(num[i],sum+num[i]); //
20        //choose if restart
21        max_sum = max(sum,max_sum); //
22        //update max subarray
23    }
24    cout << max_sum;
25    return 0;
26 }

```

7.6 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 }

```

7.2 1D1D dp

```

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 p = 1000000009;
8 LL memo[RSIZE];
9 LL cata[LL times]{
10     if(memo[times] >= 0) return memo[
11         times];
12     LL sum = 0;
13     for(LL i = 0; i < times; i++){
14         LL temp = cata(i)*cata(times-1-i)
15         %p;
16         sum = (sum+temp) % p;
17     }
18 }

```

7.5 mergesort

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define good ios_base::sync_with_stdio(0)
4 ;cin.tie(0)
5 #define NORMALSIZE 8
6 typedef long long LL;
7 void merge(LL *,LL,LL,LL); //prototype

```

7.7 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 }

```

7.8 逆序數對

```

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 LL number[100002] = {0};
8
9 LL inverse_pair(LL a[], LL left, LL right){
10     LL i, j, mid = (left+right)/2;
11     if(left+1 >= right) return 0; //one
12     num has no pair
13     LL ans = inverse_pair(a, left, mid) +
14     inverse_pair(a, mid+1, right); //
15     pair in same side
16     LL cross = 0; //pair cross both side
17     sort(a+mid, a+right); //sort right part
18     of num
19     for(i = 0; i < mid; i++){
20         cross += lower_bound(a+mid, a+
21         right, a[i]) - (a+mid); //
22         quantity of smaller than a[i]
23     }
24     return ans+cross;
25
26 int main(){
27     good;
28     LL n;
29     cin >> n;
30     for(LL i = 0; i < n; i++){
31         cin >> number[i];
32     }
33     cout << inverse_pair(number, 0, n);
34     return 0;
35 }

```

8 zformula

8.1 formula

8.1.1 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

8.1.2 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：

- $I(G) + Cv(G) = |V|$
- $M(G) + Ce(G) = |V|$

- 對於連通二分圖：

- $I(G) = Cv(G)$
- $M(G) = Ce(G)$

- 最大權閉圖：

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

- 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 $((U + |V| - flow(S, T))/2 > 0)$ $l = mid$
 else $r = mid$
- $ans = min_cut(S, T)$
- $|E| = 0$ 要特殊判斷

- 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

8.1.3 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： $O \left(V^{1/2} E \right)$

8.1.4 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
5         ,g)
6         找出一組合法x[i]使f(g)最大;
7         if(f(g)>0) l=g;
8         else r=g;
9     }
10    Ans = r;
11 }
12 Dinkelbach(){
13     g=任意狀態(通常設為0);
14     do{
15         Ans=g;
16         for(i:所有元素)D[i]=B[i]-g*C[i]; //D(i
17         ,g)
18         找出一組合法x[i]使f(g)最大;
19         p=0, q=0;
20         for(i:所有元素)
21             if(x[i])p+=B[i], q+=C[i];
22         g=p/q; //更新解，注意q=0的情況
23     }while(abs(Ans-g)>EPS);
24     return Ans;
25 }

```

8.1.5 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 $M(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$

8.1.6 基本數論

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

8.1.7 排組公式

- k 卡特蘭 $\frac{C_n^{kn}}{n(k-1)+1} \cdot C_m^n = \frac{n!}{m!(n-m)!}$
- $H(n, m) \cong x_1 + x_2 + \dots + x_n = k, num = C_k^{n+k-1}$
- 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

- $D_n = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} \dots + \frac{(-1)^n \frac{1}{n!})$
- $D_n = (n-1)(D_{n-1} + D_{n-2}), D_0 = 1, D_1 = 0$
- From $D_0 : 1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496$

- Binomial Equality

- $\sum_k \binom{r}{m+k} \binom{s}{n-k} = \binom{r+s}{m+n}$
- $\sum_k \binom{l}{m+k} \binom{s}{n+k} = \binom{l+s}{l-m+n}$
- $\sum_k \binom{l}{m+k} \binom{s}{n-k} (-1)^k = (-1)^{l+m} \binom{s-m}{n-l}$
- $\sum_{k \leq l} \binom{l-k}{m} \binom{s}{k-n} (-1)^k = (-1)^{l+m} \binom{s-m-1}{l-n-m}$
- $\sum_{0 \leq k \leq l} \binom{l-k}{m} \binom{q+k}{n} = \binom{l+q+1}{m+n+1}$
- $\binom{r}{m} \binom{m}{k} = \binom{r}{k} \binom{r-k}{m-k}$
- $\sum_{k \leq n} \binom{r+k}{k} = \binom{r+n+1}{n}$
- $\sum_{0 \leq k \leq n} \binom{k}{m} = \binom{n+1}{m+1}$
- $\sum_{k \leq m} \binom{m+r}{k} x^k y^{m-k} = \sum_{k \leq m} \binom{-r}{k} (-x)^k (x+y)^{m-k}$

8.1.8 冪次，冪次和

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

8.1.9 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 $\frac{1}{24} (3^6 + 6 \times 3^3 + 3 \times 3^4 + 8 \times 3^2 + 6 \times 3^3) = 57$

8.1.10 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) $M[i][i] = degree(V_i), M[i][j] = -1, \text{if have } E(i, j), 0 \text{ if no edge. delete any one row and col in } A, ans = det(A)$

Codebook - ss

Contents

1 Data Structure	1	3 Number Theory	3	6.4 two number(another slide)	5
1.1 01 背包	1	3.1 bit set	3	6.5 模板	6
1.2 array simulate linked list	1	3.2 matrix exponential	3		
1.3 binary search	1	3.3 SpeedExpo	4		
1.4 discretization	1	3.4 大數取模	4		
1.5 half enumeration	1	3.5 模逆元	4		
1.6 LCS	1				
1.7 LIS	1	4 String	4		
1.8 skew heap	1	4.1 linked list	4		
		4.2 manacher(最小回文字串)	4		
2 Graph	2	5 Tree Problem	4	7 other	6
2.1 bellman ford	2	5.1 findLCA	4	7.1 1D0D dp	6
2.2 BFS	2	5.2 kruskal(MST)	4	7.2 1D1D dp	6
2.3 DAG 最長最短	2	5.3 LCA	4	7.3 2D0D dp	6
2.4 DFS	2	5.4 Prim(MST)	5	7.4 maximum subarray	6
2.5 dijkstra	2	5.5 segment tree	5	7.5 mergesort	6
2.6 floyd warshall	2			7.6 sliding window	6
2.7 topology sort	3	6 default	5	7.7 WhatDay	6
		6.1 8 queen	5	7.8 逆序數對	6
		6.2 randomize	5		
		6.3 sweepline	5	8 zformula	7
				8.1 formula	7
				8.1.1 Pick 公式	7
				8.1.2 圖論	7
				8.1.3 dinic 特殊圖複雜度	7
				8.1.4 0-1 分數規劃	7
				8.1.5 學長公式	7
				8.1.6 基本數論	7
				8.1.7 排組公式	7
				8.1.8 冪次, 冪次和	7
				8.1.9 Burnside's lemma	7
				8.1.10 Count on a tree	7

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 }
```