

2015-2016 ACM-ICPC Nordic Collegiate Programming
Contest (NCPC 2015)

Chun-Hung Tseng

June 1, 2016

Problem A

```
1  #include <bits/stdc++.h>
2  // LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4  #ifdef _WIN32
5  #define lld "I64d"
6  #else
7  #define lld "lld"
8  #endif
9
10 typedef long long int ll;
11
12 using namespace std;
13
14 vector<int> g[100010];
15 int deg[100010];
16
17 struct UFDS {
18     int par[100010];
19     vector<int> m[100010];
20     void init(int n)
21     {
22         memset(par, -1, sizeof(par));
23         for (int i = 0; i < n; i++) {
24             m[i].clear();
25             m[i].push_back(i);
26         }
27     }
28     int root(int x)
29     {
30         return par[x] < 0 ? x : par[x] = root(par[x]);
31     }
32     void merge(int x, int y)
33     {
34         x = root(x);
35         y = root(y);
36
37         if (x != y) {
38             if (par[x] > par[y])
39                 swap(x, y);
40             par[x] += par[y];
41             m[x].insert(m[x].end(), m[y].begin(), m[y].end());
42             m[y].clear();
43
44             par[y] = x;
45         }
46     }
47 };
48
49 int main()
50 {
51     int n, k;
52     scanf("%d %d", &n, &k);
53     if (k == 0) {
54         printf("2\n");
55         return 0;
56     }
57
58     UFDS uf;
59     uf.init(n);
```

```

60     for (int i = 0; i < n; i++) {
61         g[i].clear();
62         deg[i] = 0;
63     }
64
65     for (int i = 0; i < k; i++) {
66         int u, v;
67         scanf("%d %d", &u, &v);
68
69         g[u].push_back(v);
70         g[v].push_back(u);
71         deg[u]++;
72         deg[v]++;
73         uf.merge(u, v);
74     }
75
76     vector<int> ans;
77     for (int i = 0; i < n; i++) {
78         if (uf.par[i] < 0) {
79             // tree center
80             queue<int> q[2];
81             int useq = 0;
82             for (int j = 0; j < (int)uf.m[i].size(); j++) {
83                 // add leaf to queue
84                 int v = uf.m[i][j];
85                 if (deg[v] == 1) {
86                     q[useq].push(v);
87                 }
88             }
89
90             int change = 0;
91             int level[100010] = {0};
92             while (q[useq].empty() == false) {
93                 int cur = q[useq].front();
94                 q[useq].pop();
95
96                 for (int j = 0; j < (int)g[cur].size(); j++) {
97                     int v = g[cur][j];
98                     deg[v]--;
99                     if (deg[v] == 1) {
100                         level[v] = level[cur] + 1;
101                         change = max(change, level[v]);
102                         q[useq ^ 1].push(v);
103                     }
104                 }
105
106                 if (q[useq].size() == 0) {
107                     useq ^= 1;
108                     if (q[useq].size() == 0) {
109                         break;
110                     }
111                 }
112             }
113             int cntCenter = 0;
114             for (int u : uf.m[i]) {
115                 // printf("u %d\n", u);
116                 if (level[u] == change)
117                     cntCenter++;
118             }
119             // printf("%d\n", cntCenter);
120             ans.push_back(2 * change + cntCenter - 1);
121         }

```

```

122     }
123     sort(ans.begin(), ans.end());
124     reverse(ans.begin(), ans.end());
125     /*
126         for(int i = 0; i < (int)ans.size(); i++)
127             printf("a %d\n", ans[i]);
128     */
129     if (ans.size() == 1)
130         printf("%d\n", ans[0]);
131     else {
132         int mx = ans[0];
133         mx = max(mx, (ans[0] + 1) / 2 + (ans[1] + 1) / 2 + 1);
134         if (ans.size() > 2)
135             mx = max(mx, (ans[1] + 1) / 2 + (ans[2] + 1) / 2 + 2);
136         printf("%d\n", mx);
137     }
138
139     return 0;
140 }

```

A/main.cpp

Problem C

```

1  #include <bits/stdc++.h>
2  // LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4  #ifdef _WIN32
5  #define ll long long
6  #else
7  #define ll long long
8  #endif
9
10 typedef long long int ll;
11
12 using namespace std;
13
14 int main()
15 {
16     char inp[1000];
17     while (scanf("%s", inp) == 1) {
18         int len = strlen(inp);
19         int ans = 0;
20         const char *str = "PER";
21         for (int i = 0; i < len / 3; i++) {
22             for (int j = 0; j < 3; j++) {
23                 if (inp[i * 3 + j] != str[j])
24                     ans++;
25             }
26         }
27         printf("%d\n", ans);
28     }
29
30     return 0;
31 }

```

C/main.cpp

Problem D

```
1  #include <bits/stdc++.h>
2  // LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4  #ifdef _WIN32
5  #define lld "I64d"
6  #else
7  #define lld "lld"
8  #endif
9
10 using namespace std;
11 typedef long long int ll;
12 typedef pair<int, int> ii;
13
14 int main()
15 {
16     int n, k;
17     while (scanf("%d %d", &n, &k) == 2) {
18         vector<ii> inp;
19         for (int i = 0; i < n; i++) {
20             int t;
21             scanf("%d", &t);
22             inp.push_back(ii(t, 1));
23             inp.push_back(ii(t + 1000, 0));
24         }
25         sort(inp.begin(), inp.end());
26
27         int cnt = 0, mx = 0;
28         for (int i = 0; i < (int)inp.size(); i++) {
29             if (inp[i].second == 1)
30                 cnt++;
31             else
32                 cnt--;
33             mx = max(mx, cnt);
34             // printf("%d %d\n", cnt, mx);
35         }
36         // printf("%d %d\n", mx / k, mx % k);
37         printf("%d\n", mx / k + (mx % k == 0 ? 0 : 1));
38     }
39
40     return 0;
41 }
```

D/main.cpp

Problem E

```
1  #include <bits/stdc++.h>
2  // LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4  #ifdef _WIN32
5  #define lld "I64d"
6  #else
7  #define lld "lld"
8  #endif
9
10 using namespace std;
11 typedef long long int ll;
```

```

12 typedef pair<int, int> ii;
13
14 ii inp[100010];
15
16 bool cmp(ii a, ii b)
17 {
18     if (a.second == b.second)
19         return a.first < b.first;
20     return a.second < b.second;
21 }
22 int main()
23 {
24     int n, k;
25     while (scanf("%d %d", &n, &k) == 2) {
26         for (int i = 0; i < n; i++) {
27             int x, y;
28             scanf("%d %d", &x, &y);
29
30             inp[i] = ii(x, y);
31         }
32         sort(inp, inp + n, cmp);
33
34         multiset<int, greater<int>> s;
35         int ans = 0;
36         for (int i = 0; i < k; i++)
37             s.insert(0);
38         // for(auto i : s)
39         // printf("s %d\n", i);
40         for (int i = 0; i < n; i++) {
41             auto it = s.lower_bound(inp[i].first);
42             // printf("%d lb %d\n", inp[i].first, it == s.end() ? -1 : *it);
43             if (it != s.end()) {
44                 s.erase(it);
45                 s.insert(inp[i].second);
46                 ans++;
47             }
48         }
49         printf("%d\n", ans);
50     }
51
52     return 0;
53 }

```

E/main.cpp

Problem G

```

1 #include <bits/stdc++.h>
2 // LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4 #ifdef _WIN32
5 #define lld "I64d"
6 #else
7 #define lld "lld"
8 #endif
9
10 using namespace std;
11
12 typedef long long int ll;
13 typedef pair<int, int> ii;

```

```

14
15 int dist(int x, int y, int a, int b)
16 {
17     int dx = x - a;
18     int dy = y - b;
19     return dx * dx + dy * dy;
20 }
21
22 typedef map<ii, int> data;
23 data loc[10010]; // for every x-coor, store location and count
24 set<ii> sloc;
25 int main()
26 {
27     int n;
28     while (scanf("%d", &n) == 1) {
29         for (int i = 0; i < n; i++) {
30             int x, y;
31             scanf("%d %d", &x, &y);
32
33             loc[x][(ii(x, y))]+=;
34         }
35
36         int k;
37         scanf("%d", &k);
38         for (int i = 0; i < k; i++) {
39             int x, y, r;
40             scanf("%d %d %d", &x, &y, &r);
41             if (sloc.find(ii(x, y)) != sloc.end())
42                 continue;
43
44             for (int j = (x - r >= 0 ? x - r : 0);
45                  j <= (x + r <= 10000 ? x + r : 10000); j++) {
46                 int dy = sqrt(r * r - (j - x) * (j - x));
47                 int uppery = y + dy, lowery = y - dy;
48
49                 auto it_begin = loc[j].lower_bound(ii(j, lowery));
50                 auto it_end = loc[j].upper_bound(ii(j, uppery));
51
52                 loc[j].erase(it_begin, it_end);
53             }
54         }
55
56         int ans = 0;
57         for (int i = 0; i < 10010; i++) {
58             for (auto j : loc[i])
59                 ans += j.second;
60         }
61         printf("%d\n", ans);
62     }
63     return 0;
64 }

```

G/main.cpp