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

Chun-Hung Tseng

June 1, 2016

Problem A

```
#include <bits/stdc++.h>
// LLONG_MIN LLONG_MAX INT_MIN INT_MAX

#ifdef _WIN32
#define lld "I64d"
#else
#define lld "lld"
#endif

typedef long long int ll;

using namespace std;

vector<int> g[100010];
int deg[100010];

struct UFDS {
    int par[100010];
    vector<int> m[100010];
    void init(int n)
    {
        memset(par, -1, sizeof(par));
        for (int i = 0; i < n; i++) {
            m[i].clear();
            m[i].push_back(i);
        }
    }
    int root(int x)
    {
        return par[x] < 0 ? x : par[x] = root(par[x]);
    }
    void merge(int x, int y)
    {
        x = root(x);
        y = root(y);

        if (x != y) {
            if (par[x] > par[y])
                swap(x, y);
            par[x] += par[y];
            m[x].insert(m[x].end(), m[y].begin(), m[y].end());
            m[y].clear();

            par[y] = x;
        }
    }
};

int main()
{
    int n, k;
    scanf("%d %d", &n, &k);
    if (k == 0) {
        printf("2\n");
        return 0;
    }

    UFDS uf;
    uf.init(n);
    for (int i = 0; i < n; i++) {
        g[i].clear();
        deg[i] = 0;
    }

    for (int i = 0; i < k; i++) {
        int u, v;
        scanf("%d %d", &u, &v);

        g[u].push_back(v);
        g[v].push_back(u);
        deg[u]++;
        deg[v]++;
        uf.merge(u, v);
    }

    vector<int> ans;
    for (int i = 0; i < n; i++) {
        if (uf.par[i] < 0) {
            // tree center
            queue<int> q[2];
            int useq = 0;
            for (int j = 0; j < (int)uf.m[i].size(); j++) {
                // add leaf to queue
                int v = uf.m[i][j];
                if (deg[v] == 1) {
                    q[useq].push(v);
                }
            }

            int change = 0;
            int level[100010] = {0};
            while (q[useq].empty() == false) {
                int cur = q[useq].front();
                q[useq].pop();

                for (int j = 0; j < (int)g[cur].size(); j++) {
                    int v = g[cur][j];
                    deg[v]--;
                    if (deg[v] == 1) {
                        level[v] = level[cur] + 1;
                        change = max(change, level[v]);
                        q[useq ^ 1].push(v);
                    }
                }

                if (q[useq].size() == 0) {
                    useq ^= 1;
                    if (q[useq].size() == 0) {
                        break;
                    }
                }
            }
            int cntCenter = 0;
            for (int u : uf.m[i]) {
                // printf("u %d\n", u);
                if (level[u] == change)
                    cntCenter++;
            }
            // printf("%d\n", cntCenter);
            ans.push_back(2 * change + cntCenter - 1);
        }
    }
    sort(ans.begin(), ans.end());
    reverse(ans.begin(), ans.end());
    /*
       for(int i = 0; i < (int)ans.size(); i++)
       printf("a %d\n", ans[i]);
    */
    if (ans.size() == 1)
        printf("%d\n", ans[0]);
    else {
        int mx = ans[0];
        mx = max(mx, (ans[0] + 1) / 2 + (ans[1] + 1) / 2 + 1);
        if (ans.size() > 2)
            mx = max(mx, (ans[1] + 1) / 2 + (ans[2] + 1) / 2 + 2);
        printf("%d\n", mx);
    }

    return 0;
}
```

Problem C

```
#include <bits/stdc++.h>
// LLONG_MIN LLONG_MAX INT_MIN INT_MAX

#ifdef _WIN32
#define lld "I64d"
#else
#define lld "lld"
#endif

typedef long long int ll;

using namespace std;

int main()
{
    char inp[1000];
    while (scanf("%s", inp) == 1) {
        int len = strlen(inp);
        int ans = 0;
        const char *str = "PER";
        for (int i = 0; i < len / 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (inp[i * 3 + j] != str[j])
                    ans++;
            }
        }
        printf("%d\n", ans);
    }

    return 0;
}
```

Problem D

```
#include <bits/stdc++.h>
// LLONG_MIN LLONG_MAX INT_MIN INT_MAX

#ifdef _WIN32
```

```
#define lld "I64d"
#else
#define lld "lld"
#endif

using namespace std;
typedef long long int ll;
typedef pair<int, int> ii;

int main()
{
    int n, k;
    while (scanf("%d %d", &n, &k) == 2) {
        vector<ii> inp;
        for (int i = 0; i < n; i++) {
            int t;
            scanf("%d", &t);
            inp.push_back(ii(t, 1));
            inp.push_back(ii(t + 1000, 0));
        }
        sort(inp.begin(), inp.end());

        int cnt = 0, mx = 0;
        for (int i = 0; i < (int)inp.size(); i++) {
            if (inp[i].second == 1)
                cnt++;
            else
                cnt--;
            mx = max(mx, cnt);
            // printf("%d %d\n", cnt, mx);
        }
        // printf("%d %d\n", mx / k, mx % k);
        printf("%d\n", mx / k + (mx % k == 0 ? 0 : 1));
    }

    return 0;
}
```

Problem E

```
#include <bits/stdc++.h>
// LLONG_MIN LLONG_MAX INT_MIN INT_MAX

#ifdef _WIN32
#define lld "I64d"
#else
#define lld "lld"
#endif

using namespace std;
typedef long long int ll;
typedef pair<int, int> ii;

ii inp[100010];

bool cmp(ii a, ii b)
{
    if (a.second == b.second)
        return a.first < b.first;
    return a.second < b.second;
}
```

```
int main()
{
    int n, k;
    while (scanf("%d %d", &n, &k) == 2) {
        for (int i = 0; i < n; i++) {
            int x, y;
            scanf("%d %d", &x, &y);

            inp[i] = ii(x, y);
        }
        sort(inp, inp + n, cmp);

        multiset<int, greater<int>> s;
        int ans = 0;
        for (int i = 0; i < k; i++)
            s.insert(0);
        // for(auto i : s)
        // printf("s %d\n", i);
        for (int i = 0; i < n; i++) {
            auto it = s.lower_bound(inp[i].first);
            // printf("%d lb %d\n", inp[i].first, it == s.end() ? -1 : *it);
            if (it != s.end()) {
                s.erase(it);
                s.insert(inp[i].second);
                ans++;
            }
        }
        printf("%d\n", ans);
    }

    return 0;
}
```

Problem G

```
#include <bits/stdc++.h>
// LLONG_MIN LLONG_MAX INT_MIN INT_MAX

#ifdef _WIN32
#define lld "I64d"
#else
#define lld "lld"
#endif

using namespace std;

typedef long long int ll;
typedef pair<int, int> ii;

int dist(int x, int y, int a, int b)
{
    int dx = x - a;
    int dy = y - b;
    return dx * dx + dy * dy;
}

typedef map<ii, int> data;
data loc[10010]; // for every x-coor, store location and count
set<ii> sloc;
int main()
{
    int n;
    while (scanf("%d", &n) == 1) {
        for (int i = 0; i < n; i++) {
            int x, y;
            scanf("%d %d", &x, &y);

            loc[x][(ii(x, y))]++;
        }

        int k;
        scanf("%d", &k);
        for (int i = 0; i < k; i++) {
            int x, y, r;
            scanf("%d %d %d", &x, &y, &r);
            if (sloc.find(ii(x, y)) != sloc.end())
                continue;

            for (int j = (x - r >= 0 ? x - r : 0);
                 j <= (x + r <= 10000 ? x + r : 10000); j++) {
                int dy = sqrt(r * r - (j - x) * (j - x));
                int uppery = y + dy, lowery = y - dy;

                auto it_begin = loc[j].lower_bound(ii(j, lowery));
                auto it_end = loc[j].upper_bound(ii(j, uppery));

                loc[j].erase(it_begin, it_end);
            }
        }

        int ans = 0;
        for (int i = 0; i < 10010; i++) {
            for (auto j : loc[i])
                ans += j.second;
        }
        printf("%d\n", ans);
    }
    return 0;
}
```