HackerX ★

Problem Setter: Wanbo

Problem Tester: Cao Peng, Khongor

Approach

What's the requirement if one hackerX missiles can defend both (ti, fi) and (tj, fj) when ti <= tj?

tj - ti >= \|fi - fj\| (because you just need \|fi - fj\| units time to change the frequency from fi to fj, so you can defend tj after defend ti by this hackerX missiles if you need. So if we regard every coming missiles (ti, fi) as an vertex i, and if tj - ti >= \|fi - fj\| then we add an edge from i to j, this edge indicates that one hackerX can defend the jth missile after defending the ith missile.

So the solution for this challenge seems obvious, we just need to find the "Minimum Path Cover"(Can be changed to MaxFlow problem) of this graph.

But n can be as large as 100000, and there are at most O(n^2) edges, even generating the graph will get TLE, we need to optimize our solution.

We haven't taken advantage of the particularity of this graph in the previous analysis, we just look this as a more general problem.

tj - ti >= \|fi - fj\| <==> If fi < fj, tj - ti >= fj - fi ==> ti - fi <= tj - fj ==> Ai <= Aj (Ai = ti - fi) If fi >= fj, tj - ti >= fi - fj ==> ti + fi <= tj + fj ==> Bi <= Bj (Bi = ti + fi)

Can we remove some constraints?

1. ti <= tj
2. If fi < fj, Ai <= Aj ==> Ai + 2 * fi <= Aj + 2 * fj ==> Bi <= Bj
3. If fi >= fj, Bi <= Bj ==> Bi - 2 * fi <= Bj - 2 * fj ==> Ai <= Aj

We can combine 2) with 3) to "Ai <= Aj && Bi <= Bj" Ai + Bi <= Aj + Bj ==> ti <= tj ==> ti <= tj can be removed.

==>

edge(i->j) <==> Ai <= Aj && Bi <= Bj

We need to find the minimum path cover in the graph which edge(i->j) <==> Ai <= Aj && Bi <= Bj.

According to Dilworth's theorem, the minimum path cover in this graph is equal to the longest anti-chain. So we just need to choose a maximum vertexes subset from the graph, where any of two do not have an edge. 1. Sort the (Ai, Bi) by the first key value Ai, if Ai == Aj, then smaller Bi comes first. 2. The longest decreasing subsequence of Bi will be the answer.(This is a very classic question that can be solved by O(nlgn).

Problem Setter's code:

C++

```
#include <map>
#include <set>
#include <list>
#include <queue>
#include <deque>
#include <stack>
#include <bitset>
#include <vector>
#include <ctime>
#include <cmath>
#include <cstdio>
#include <string>
#include <cstring>
#include <cassert>
#include <numeric>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <iostream>
#include <algorithm>
using namespace std;
```

```
typedef long long       LL;
typedef pair<int, int>  PII;
typedef pair<LL, LL>    PLL;
typedef vector<int>     VI;
typedef vector<LL>      VL;
typedef vector<PII>     VPII;
typedef vector<PLL>     VPLL;
#define MM(a,x) memset(a,x,sizeof(a));
#define ALL(x)  (x).begin(), (x).end()
#define P(x)    cerr<<"["#x<<" = "<<(x)<<"]\n"
#define PP(x,i)    cerr<<"["#x<<i<<" = "<<x[i]<<"]\n"
#define P2(x,y)    cerr<<"["#x" = "<<(x)<<", "#y" = "<<(y)<<"]\n"
#define TM(a,b)    cerr<<"["#a" -> "#b": "<<1e3*(b-a)/CLOCKS_PER_SEC<<"ms]\n";
#define FOR(it,v) for(__typeof(v.begin()) it=v.begin();it!=v.end();it++)
#define rep(i, n) for(int i = 0; i < n; i++)
#define UN(v) sort(ALL(v)), v.resize(unique(ALL(v))-v.begin())
#define mp make_pair
#define pb push_back
#define x first
#define y second
struct _ {_() {ios_base::sync_with_stdio(0);}} _;
template<class T> void PV(T a, T b) {while(a != b)cout << *a++, cout << (a != b ? " " : "\n");}
template<class T> inline bool chmin(T &a, T b) {return a > b ? a = b, 1 : 0;}
template<class T> inline bool chmax(T &a, T b) {return a < b ? a = b, 1 : 0;}
template<class T> string tostring(T x, int len = 0) {stringstream ss; ss << x; string r = ss.str(); if(r.length() < len) r =
template<class T> void convert(string x, T& r) {stringstream ss(x); ss >> r;}
template<class A, class B> ostream& operator<<(ostream &o, pair<A, B> t) {o << "(" << t.x << ", " << t.y << ")"; return o;}
const int inf = 0x3f3f3f3f;
const int mod = int(1e9) + 7;
const int N = 111111;

int t[N], f[N];
int d[N];
int n;


int LIS(VI v) {
    fill(d, d + n, 1);
    for(int i = 0; i < n; i++)
        for(int j = 0; j < i; j++)
            if(v[i] > v[j]) chmax(d[i], d[j] + 1);
    int res = 0;
    for(int i = 0; i < n; i++) chmax(res, d[i]);
    return res;
}

int LIS1(VI v) {
    MM(d, 0x3f);
    d[1] = v[0];
    for(int i = 1; i < v.size(); i++) {
        int t = upper_bound(d + 1, d + N, v[i]) - d;
        if(t != 1 && d[t - 1] >= v[i]) continue;
        d[t] = v[i];
    }
    int res = 0;
    for(int i = 1; i < N; i++) if(d[i] != inf) res = i;
    return res;
}

int main() {
    cin >> n;
    for(int i = 0; i < n; i++) cin >> t[i] >> f[i];
    for(int i = 1; i < n; i++) assert(t[i] >= t[i - 1]);
    VI v;
    VPII vp;
    for(int i = 0; i < n; i++) vp.pb(mp(t[i] - f[i], t[i] + f[i]));
    sort(ALL(vp));
    for(int i = 0; i < n; i++) v.pb(vp[i].second);
    reverse(ALL(v));
    //PV(ALL(v));
    cout << LIS1(v) << endl;
    //P2(LIS1(v), LIS(v));
    //assert(LIS1(v) == LIS(v));
    return 0;
}
```

Problem Tester's code:

C++

```
#include <cstdio>
#include <algorithm>
#include <set>

using namespace std;

pair<int,int> a[100005];
set<int> have;

int main() {
int n;
    scanf("%d",&n);
    for (int i = 0; i < n; ++i)  {
        int x,y;
        scanf("%d%d",&x,&y);
        a[i] = make_pair(x + y,x - y);
    }
    sort(a, a + n);
    for (int i = 0; i < n; ++i) {
        //printf("%d %d\n",a[i].first,a[i].second);

        set<int>::iterator t =  have.lower_bound(a[i].second);

        if ((t != have.end()) && (*t == a[i].second)) {
            continue;
        }
        if (t != have.begin()) {
            have.erase(--t);

        }
        have.insert(a[i].second);
    }
    printf("%d\n",have.size());
    return 0;
}
```

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
#include <utility>
using namespace std;

#define Pair pair<int, int>

#define MAX 100000

Pair v[MAX];
int a[MAX];
int dp[MAX + 1];

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        int x, y;
        scanf("%d%d", &x, &y);
        v[i] = make_pair(x - y, x + y);
    }
    sort(v, v + n);
    for (int i = 0; i < n; i++)
        a[i] = v[n - i - 1].second;
    int r = 1;
    dp[1] = a[0];
    for (int i = 1; i < n; i++) {
        int low = 1, high = r;
        if (a[i] <= dp[1]) {
            dp[1] = a[i];
            continue;
        }
        while (low < high) {
            int mid = (low + high + 1) / 2;
            if (dp[mid] >= a[i])
                high = mid - 1;
            else
                low = mid;
        }
        if (low == r) r++;
        dp[low + 1] = a[i];
    }
    cout << r << endl;
    return 0;
}
```

Feedback

Was this editorial helpful?

Yes No