Paradigmas de programação

Pares de pontos mais próximos

```
const double EPS = 1e-9;
const double INF = 1e18;
struct point{
    double x;
    double y;

    point() { x=y=0.0; }
    point(double _x, double _y): x(_x), y(_y) {}

    bool operator <(point other) const{
        return fabs(x-other.x)<EPS ? y<other.y : x<other.x;
    }
};
double dis(point a, point b){
    return hypot(a.x-b.x, a.y-b.y);
}

pair<point, point> closet_pair(vector<point>& vs){
    sort(vs.begin(), vs.end());
    int n = vs.size();
    set<point> st;
    double d = INF;
    auto closest = make_pair(point(), point());
    st.insert(point(vs[0].y, vs[0].x));
    for(int i=1; i<n; ++i){
        auto p = vs[i];
        auto it = st.lower_bound(point(p.y-d, -INF));
        while(it != st.end()){
            auto q = point(it->y, it->x);
            if(q.x < p.x-d){
                it = st.erase(it);
                continue;
            }
            else if(q.y > p.y+d) break;
            auto t = dis(p, q);
            if(t<d){
                d = t;
                closest = make_pair(p, q);
            }
            ++it;
        }
        st.insert(point(p.y, p.x));
    }
    return closest;
}
```