```cpp
#include "../bits/stdc++.h"
/**
 * kD-tree (k = 2)
 * k次元二分探索木(?)
 * 構築までしか書いてない
 */
template <typename T>
class KdTree
{
    static const int Dim = 2;
    struct Node
    {
        int idx; // 元々の頂点番号
        Node *ch[2];
        int axis;

        Node() : idx(-1), axis(-1) { ch[0] = ch[1] = nullptr; }
    };

    Node *root;
    std::vector<T> points;

    // 構築
    // indices := 点の元々のインデックス
    Node *build(int *indices, int pointSize, int depth)
    {
        if (pointSize <= 0)
            return nullptr;

        int axis = depth % Dim;
        int mid = (pointSize - 1) / 2;

        std::nth_element(indices, indices + mid, indices + pointSize, [&](int lhs, int rhs) {
            return points[lhs][axis] < points[rhs][axis];
        });

        Node *node = new Node();
        node->idx = indices[mid];
        node->axis = axis;

        node->ch[0] = build(indices, mid, depth + 1);
        node->ch[1] = build(indices + mid + 1, pointSize - mid - 1, depth + 1);

        return node;
    }
};

struct Point
{
    static const int Dim = 2;
    std::vector<double> ps;
    Point() {}
    Point(double x, double y) : ps(2)
    {
        ps[0] = x;
        ps[1] = y;
    }

    double operator[](std::size_t t)
    {
        assert(0 <= t && t <= 1);
        return ps[t];
    }
};


```