

빌딩 번호 (520)

스타링크

최백준 choi@startlink.io



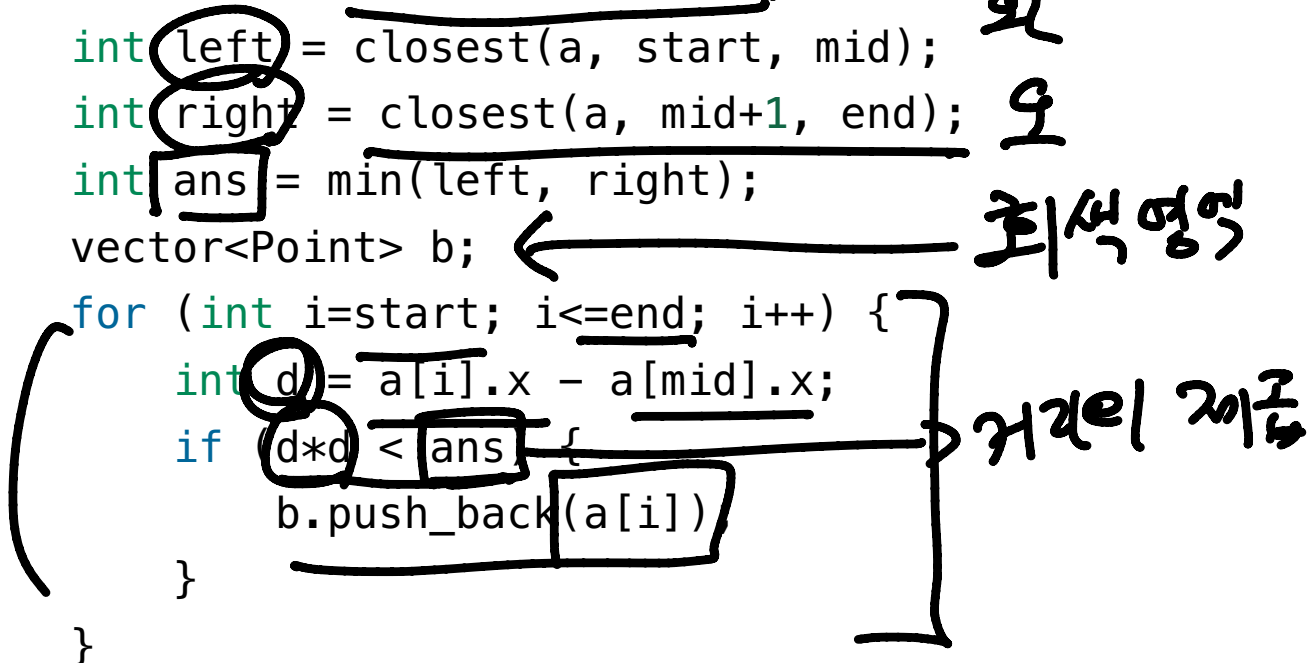
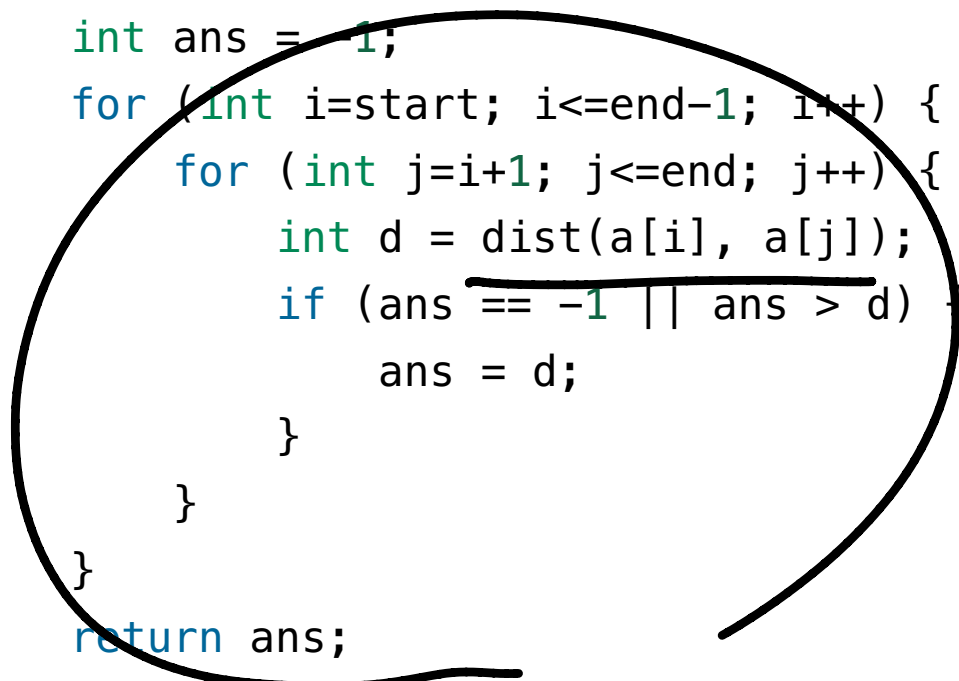
C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6 using Result = vector<pair<int,int>>;
7 struct Building {
8     int left, right, height;
9 };
10 void append(Result &ans, pair<int,int> p) {
11     int n = ans.size();
12     if (n > 0) {
13         if (ans[n-1].second == p.second) {
14             return;
15         }
16         if (ans[n-1].first == p.first) {
17             ans[n-1].second = p.second;
18             return;
19         }
20     }
21     ans.push_back(p);
22 }
23 Result merge(Result &l, Result &r) {
24     Result ans;
25     int h1 = 0;
26     int h2 = 0;
27     int i = 0;
28     int j = 0;
29     while (i < l.size() && j < r.size()) {
30         auto &u = l[i];
31         auto &v = r[j];
32         if (u.first < v.first) {
33             int x = u.first;
34             h1 = u.second;
35             int h = max(h1, h2);
36             append(ans, make_pair(x, h));
37             i += 1;
38         } else {
39             int x = v.first;
40             h2 = v.second;
41             int h = max(h1, h2);
42             append(ans, make_pair(x, h));
43             j += 1;
44         }
45     }
46     while (i < l.size()) {
47         ans.push_back(l[i]);
48         i += 1;
49     }
50     while (j < r.size()) {
51         ans.push_back(r[j]);
52         j += 1;
53     }
54     return ans;
55 }
56 Result go(vector<Building> &a, int start, int end) {
57     if (start == end) {
58         Result ans = {
59             {a[start].left, a[start].height},
60             {a[start].right, 0}
61         };
62         return ans;
63     }
64     int mid = (start + end) / 2;
65     auto l = go(a, start, mid);
66     auto r = go(a, mid+1, end);
67     return merge(l, r);
68 }
69 int main() {
70     ios_base::sync_with_stdio(false);
71     cin.tie(nullptr);
72     int n;
73     cin >> n;
74     vector<Building> a(n);
75     for (int i=0; i<n; i++) {
76         cin >> a[i].left >> a[i].height >> a[i].right;
77     }
78     sort(a.begin(), a.end(), [](auto &u, auto &v) {
79         return make_tuple(u.left, u.height, u.right) < make_tuple(v.left, v.height, v.right);
80     });
81     auto ans = go(a, 0, n-1);
82     for (auto &p : ans) {
83         cout << p.first << ' ' << p.second << ' ';
84     }
85     cout << '\n';
86     return 0;
87 }
88
```

결과	메모리	시간	코드 길이
맞았습니다!!	3164 KB	68 ms	2088 B

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4 #include <algorithm>
5 using namespace std;
6 struct Point {
7     int x, y;
8 };
9 bool cmp_x(const Point &u, const Point &v) {
10     return make_pair(u.x, u.y) < make_pair(v.x, v.y);
11 }
12 bool cmp_y(const Point &u, const Point &v) {
13     return make_pair(u.y, u.x) < make_pair(v.y, v.x);
14 }
15 int dist(Point p1, Point p2) {
16     return (p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y);
17 }
18 int brute_force(vector<Point> &a, int start, int end) {
19     int ans = -1;
20     for (int i=start; i<=end-1; i++) {
21         for (int j=i+1; j<=end; j++) {
22             int d = dist(a[i], a[j]);
23             if (ans == -1 || ans > d) {
24                 ans = d;
25             }
26         }
27     }
28     return ans;
29 }
30 int closest(vector<Point> &a, int start, int end) {
31     int n = end-start+1;
32     if (n <= 3) {
33         return brute_force(a, start, end);
34     }
35     int mid = (start + end) / 2;
36     int left = closest(a, start, mid);
37     int right = closest(a, mid+1, end);
38     int ans = min(left, right);
39     vector<Point> b;
40     for (int i=start; i<=end; i++) {
41         int d = a[i].x - a[mid].x;
42         if (d*d < ans) {
43             b.push_back(a[i]);
44         }
45     }
46     sort(b.begin(), b.end(), cmp_y);
47     int m = b.size();
48     for (int i=0; i<m-1; i++) {
49         for (int j=i+1; j<m; j++) {
50             int y = b[j].y - b[i].y;
51             if (y*y < ans) {
52                 int d = dist(b[i], b[j]);
53                 if (d < ans) {
54                     ans = d;
55                 }
56             } else {
57                 break;
58             }
59         }
60     }
61     return ans;
62 }
63 int main() {
64     ios_base::sync_with_stdio(false);
65     cin.tie(nullptr);
66     int n;
67     cin >> n;
68     vector<Point> a(n);
69     for (int i=0; i<n; i++) {
70         int x, y;
71         cin >> x >> y;
72         a[i] = {x, y};
73     }
74     sort(a.begin(), a.end(), cmp_x);
75     cout << closest(a, 0, n-1) << '\n';
76     return 0;
77 }
```



N^2N

최대 1000

break; 33

C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4 #include <algorithm>
5 using namespace std;
6 struct Point {
7     int x, y;
8 };
9 bool cmp_x(const Point &u, const Point &v) {
10     return make_pair(u.x, u.y) < make_pair(v.x, v.y);
11 }
12 bool cmp_xe(const Point &u, const Point &v) {
13     return make_pair(u.x, u.y) <= make_pair(v.x, v.y);
14 }
15 bool cmp_y(const Point &u, const Point &v) {
16     return make_pair(u.y, u.x) < make_pair(v.y, v.x);
17 }
18 int dist(Point p1, Point p2) {
19     return (p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y);
20 }
21 int brute_force(vector<Point> &a, int start, int end) {
22     int ans = -1;
23     for (int i=start; i<=end-1; i++) {
24         for (int j=i+1; j<=end; j++) {
25             int d = dist(a[i], a[j]);
26             if (ans == -1 || ans > d) {
27                 ans = d;
28             }
29         }
30     }
31     return ans;
32 }
33 int closest(vector<Point> &ax, vector<Point> &ay, int start, int end) {
34     int n = end-start+1;
35     if (n <= 3) {
36         return brute_force(ax, start, end);
37     }
38     int mid = (start + end) / 2;
39     Point mid_p = ax[mid];
40     vector<Point> ayl, ayr;
41     for (Point p : ay) {
42         if (cmp_xe(p, mid_p)) {
43             ayl.push_back(p);
44         } else {
45             ayr.push_back(p);
46         }
47     }
48     int left = closest(ax, ayl, start, mid);
49     int right = closest(ax, ayr, mid+1, end);
50     int ans = min(left, right);
51     vector<Point> b;
52     for (Point p : ay) {
53         int d = p.x - mid_p.x;
54         if (d*d < ans) {
55             b.push_back(p);
56         }
57     }
58     int m = b.size();
59     for (int i=0; i<m-1; i++) {
60         for (int j=i+1; j<m; j++) {
61             int y = b[j].y - b[i].y;
62             if (y*y < ans) {
63                 int d = dist(b[i], b[j]);
64                 if (d < ans) {
65                     ans = d;
66                 }
67             } else {
68                 break;
69             }
70         }
71     }
72     return ans;
73 }
74 int main() {
75     ios_base::sync_with_stdio(false);
76     cin.tie(nullptr);
77     int n;
78     cin >> n;
79     vector<Point> ax(n);
80     vector<Point> ay(n);
81     for (int i=0; i<n; i++) {
82         int x, y;
83         cin >> x >> y;
84         ax[i] = {x, y};
85         ay[i] = {x, y};
86     }
87     sort(ax.begin(), ax.end(), cmp_x);
88     sort(ay.begin(), ay.end(), cmp_y);
89     cout << closest(ax, ay, 0, n-1) << '\n';
90     return 0;
91 }
92
```

결과	메모리	시간	코드 길이
맞았습니다!!	21152 KB	84 ms	2283 B

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.