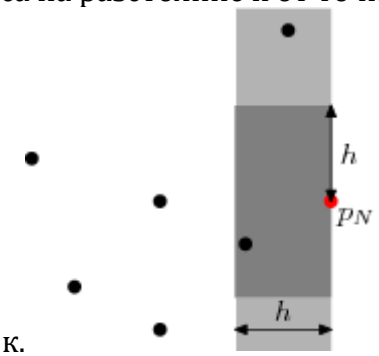
Най-близка двойка

В равнината са дадени N (2 < N ≤ 25 000) точки с целочислени координати в интервала [-10 000; 10 000]. Да се намери разстоянието между най-близките две измежду тях.

Жокер: Да предположим, че до момента сме обработили точките от 1 до N-1 (предварително сортирани по техните x координати) и сме намерили някакво текущо най-късо разстояние h. Разглеждайки точка N, търсим друга на разстояние по-малко от h (float). Поддържаме множество от вече разгледаните точки, които са на разстояние h от точка N, както е показано в светлосивия



правоъгълник. Когато преминем на следваща точка или намерим точка на разстояние по-малко от h, премахваме кандидатите от множеството. За тази цел добавяйте отзад всяка поредна точка в една помощна опашка с двойки (std::pair <int, int>) координати (X, Y). За всяка поредна точка (Xcurr, Ycurr) вадете от началото на опашката точките докато Xqueue_front <= Xcurr - h и ги изтрийте от множеството. Тъй като кандидат точките в нашето множество трябва да са сортирани по y, и в същото време да можем с логаритмична сложност да премахваме и/или добавяме към него, то идеална структура за целта е std::set. Елементите на нашето множество са двойки от тип std::pair<int, int> представляващи координатите (Y, X) на нашите кандидат точки. Забележете че координатите са разменени (в опашката не са, но ако ви е по удобно може и в опашката да ги размените), за да могат точките вътрешно да се сортират по Y, а не по X. Търсейки точка на разстояние по-малко от h от текущата N се оказва, че има смисъл да търсим само измежду точките с Y координати в тъмно сивия правоъгълник. За да открием началото на интервала от точки с Y > Ycurr - h правим двоично търсене в нашето множество чрез метода std::set::lower_bound или std::set::upper_bound. Ако "std::set <std::pair <int, int> s;", то метода ще извикаме така: auto it = s.lower_bound({int(Ycurr - h), MAX_X_PLUS_1 = 10001}). Така ще получим итератор към първата такава точка. След това ще ги итерираме, чрез it, докато it->first < Ycurr + h. Задачата е добре известна

алгоритмична задача, позната като “Closest pair problem” и за решаването ѝ съществуват няколко алгоритъма, решаващи я за O(NlogN). (https://www.topcoder.com/thrive/articles/Line%20Sweep%20Algorithms)

Първоначалната стойност на h е разстоянието между двете точки с най-малки X координати (след като сме сортирали всички точки по X), които от своя страна са и първоначалните два члена (с разменени X и Y координати) на текущото временно множество представено чрез нашия std::set. Координатите в нормален ред (не е задължително - въпрос на ваша си конвенция) на тези първи две точки вкарваме и в нашата помощна опашка.

Input Format

Входът се състои от множество тестови примери, като всеки нов ред от поредния тестов пример съдържа по две числа – координатите на поредната точка (x, y). За край на поредния тестов пример ще считаме ред, в който има само едно число - числото -1. Това, разбира се означава, че измежду зададените ви точки няма да има нито една с x координата равна на -1.

Constraints

$2 < N \leq 25\,000$

Output Format

За всеки тестов пример, трябва да изведете търсеното минимално разстояние с точност 5 знака след десетичната точка. Броят на редовете в изхода трябва да бъде равен на броя на тестовите примери.

Sample Input 0

```
0 1
2 4
6 8
7 0
-1
7 9
12 5
90 3
-1
```

Sample Output 0

```
3.60555
6.40312
```