

Задание: выбрать три разные точки заданного на плоскости множества точек, составляющие треугольник наибольшего периметра.

Проблема: очевидное решение — перебор всех существующих троек, подсчёт расстояния между каждой парой и вычисления наибольшей суммы (периметра) имеет алгоритмическую сложность $O(N^3)$ и константное время выполнения, то есть не зависит от расположения точек на множестве. Данное решение не является эффективным по времени и мною был реализован алгоритм, показывающий на стандартных входных данных (биномиальное распределение точек по каждой из координат) значительно более эффективную работу и имеющий на них сложность $O(hn+h^3)$, где $h \ll n$, то есть его сложность является квазилинейной (строгое доказательство этого утверждения отсутствует, так как выходит далеко за пределы поставленной задачи).

Описание программы: программа написана на языке программирования Java, имеет текстовый интерфейс и графический вывод решения. Программа работает с CSV-файлами и использует библиотеку `opencsv` (создатель Glen Smith, расположена на `opencsv.sourceforge.net`). Пользователь имеет возможность использовать собственный файл или сгенерировать множество прямо в программе.

Описание алгоритма: для начала работы программе необходимо считать множество из файла. У пользователя запрашивается необходимость считать или перезаписать файл и его адрес (внимание, программа не создаёт новый файл, поэтому для первого запуска необходим хотя бы пустой документ с расширением `.csv`). При необходимости генерации множества запрашивается его размер. Эти действия производятся в классе `CSVHandler` и, как уже было сказано, точки на множестве имеют биномиальное распределение (действия делегируются методами `createNewSet` и `getBinomial`). После возможного создания множества происходит чтение из файла (метод `readFile`). Каждая строка CSV-документа парсится и ей создаётся соответствующее значение класса `dot`, содержащее координаты по X и Y.

После чтения множества (`dots`) вызывается метод `findHull`, который реализует поиск выпуклой оболочки (наименьшего множества принадлежащего `dots`, содержащее внутри `dots`, см. ru.wikipedia.org/wiki/Выпуклая_оболочка) по методу Джарвиса (ru.wikipedia.org/wiki/Алгоритм_Джарвиса).

Псевдокод:

```
jarvis(S)
    pointOnHull = leftmost point in S
    i = 0
    repeat
        P[i] = pointOnHull        endpoint = S[0]
        if (endpoint == pointOnHull) or (S[j] is on left of line from P[i] to
        endpoint)                  endpoint = S[j]
        i = i+1                    pointOnHull = endpoint
    until endpoint == P[0]
```

Затем по полученному в данном результате данной операции множеству (`hull`) проводится поиск перебором троек. Очевидно, что треугольник с наибольшим периметром будет состоять из точек выпуклой оболочки.

Результат работы программы выводится на экран.

