Задачи по геометрии, математика.

Здесь кратко описаны идеи решения. Как правило, все сводится к вычислению расстояний, углов, взаимных положений точек, отрезков, прямых и т.д.

Углы между векторами можно найти из скалярных и векторных произведений. Помимо арксинуса и арккосинуса с практической точки зрения очень удобна также функция atan2 (см. в справочниках)

Часто бывает полезен "ориентированный угол зрения" — угол, под которым данный объект (например, отрезок) виден из данной точки. Он вычисляется по векторному произведению векторов и зависит от ориентации — какой вектор первый, а какой второй.

В наших задачах мы не различаем точки и векторы так как математически и точка, и вектор представляются парой координат.

Задачки из списка.

1. Самопересечение ломаной. Ломаная — массив точек.

Как проверить пересечение двух отрезков:

— концы каждого отрезка лежат по разные стороны от прямой другого отрезка. вершины - отрезки сторон - проверка пересечения отрезков ($O(n^2)$).

ВАЖНО !!! Вычислительная погрешность -> совпадение точек проверяется/фиксируется с некоторой точностью.

Подробно рассмотрено в тексте про программирование.

2. Выпуклость многоугольника. Многоугольник — массив точек с заданным направлением обхода.

— проверка, что весь лежит с одной стороны от каждого ребра — $O(n^2)$.

— проверка, что все внутренние углы имеют один знак — $O(n)$.

ВАЖНО !!! устойчивость алгоритма — как реагирует на погрешности

первый - устойчивый, второй — менее устойчивый

3. Под каким углом виден многоугольник.

— минимальный и максимальный угол на вершину (много частных случаев).

— сумма положительных и отрицательных углов видимости сторон из некоторой точки.

4. Где точка относительно многоугольника — внутри, на границе, вне.

— четность количества пересечений луча с границей — много тонких моментов — усложнение

— суммарный угол видимости сторон — 2π внутри, 0 снаружи — устойчиво и $O(n)$.

5. Минимальный и максимальный диаметры (габариты).

Минимальная ширина щели, в которую может "протиснуться" этот многоугольник.

Минимальная ширина щели, которая еще "касается" этого многоугольника.

Для максимального — расстояние между вершинами (max).

для минимального — расстояние от вершины до стороны (min max).

6. Минимальный охватывающий прямоугольник.

Если прямоугольник проходит своей стороной по стороне многоугольника, то он строится однозначно. Строим для каждой стороны и выбираем минимальный по пло-

щади. Если не так, то на каждой стороне прямоугольника лежит по одной вершине многоугольника. Можно поворачивать прямоугольник, не отрывая его от этих вершин и смотреть как меняется площадь.

7. Покрытие отрезками другого отрезка на прямой.

Упорядочиваем по первой координате все отрезки кроме первого. Потом собираем их связные объединения и смотрим как они соотносятся с первым отрезком.

8. Объединение отрезков наибольшей длины.

Упорядочиваем по первой координате. Потом собираем связные объединения.

9. Расстояние между многоугольниками.

По определению — считаем расстояние от вершин одного до сторон другого и наоборот и выбираем минимум.

10. Расстояние между выпуклыми многоугольниками за $O(n+m)$.

Выбираем две произвольные точки и потом запускаем процесс “попеременного сближения”, пока он не остановится.

11. Сборка многоугольника из отрезков.

Ищем вершины, совпалающие с заданной точностью и выстраиваем связные цепочки из отрезков. Если какая-либо цепочка замыкается сама на себя, то она и образует многоугольник.

12. Клипирование отрезков по прямоугольнику.

Отсечение по x, потом отсечение по y сторонами данного ограничивающего прямоугольника.

13. Минимальное покрывающее дерево.

Сначала каждая точка является элементом дерева. Вычисляем расстояния между точками различных элементов и объедияем элементы по наименьшему ребру. Когда останется один объединенный элемент — это и есть дерево.

14. Растущие круги на плоскости.

Моделирование процесса (последовательно набора состояний этих кругов). Состояние круга — радиус и расстояние до ближайшего (и признак — еще растет или уже остановился). Вычисляется наименьший прирост, когда сталкиваются два ближайших круга. отом все круги уведичивают диаметр на этот прирост (кроме тех, которые уже столкнулись и не растут).

15. Минимальный охватывающий круг.

Сжатие “охватывающего” кольца. Сначала кольцо фиксируется на одной точке, потом, не отрываясь от первой точки, сжимается до фиксации на второй точке. Далее, не отрываясь от двух точек, сжимается до фиксации на третьей точке. Если получился остроугольный треугольник, то конец. Если тупоугольный, то отрываем тупоугольную вершину и продолжает поиск третьей точки. Также возможно решение с касанием по диаметру.

16. Выпуклая оболочка.

Классическая задача. Алгоритмы везде описаны. Как пример:

Алгоритм оборачивания по углу.

Алгоритм оборачивания сверху и снизу.

Алгоритм "разделяй и властвуй"

17. Раздувание или сжатие выпуклого многоугоьльника.

Наружу — просто сдвиг вершин по биссектриссам углов.

Внутрь — тоже сдвиг по биссектриссам, но с проверкой более раннего пересечения биссектрисс.

18. Сечение многоугольника.

Классическая задача. Можно по-простому, с аккуратным рассмотрением случаев прохода сечения через вершину или по имеющейся стороне.

19. Лабиринт.

Заполним поле лабиринта квадратными клетками, имеющими стенки с 4 сторон. Последовательно убирая стенки между клетками и переходя в соседнюю свободную клетку в случайном направлении, можно построить частичный путь (не возвращаемся на уже пройденные клетки, и отмечаем клетки пути одним номером). Ведем путь от входной клетки пока это удается (рядом с очередной помеченной клеткой есть другая свободная, в которую можно перейти). Когда из очередной клетки идти некуда, выбираем одну из предыдущих клеток, из которых еще можно сделать переход на свободную клетку, и строим путь туда. В конце концов все свободные клетки присоединятся к текущему пути (и в том числе выход из лабиринта). Можно ограничивать количество шагов в каждом частичном пути, тогла лабиринт получится более разветвленным.

20. Пересечение выпуклых многоугольников.

Классическая задача. Один их вариантов — провести вертикальные прямые через все вершины обоих многоугольников. Эти линии разобьют каждый многугольник на трапеции (или треугольники как вырожденный случай трапеции). Теперь задача свобится к построению пересечения трапеций с основаниями, лежащими на этих прямых. А это перебор нескольких случаев. Потом убираются лишние вершины, лежащие на одной прямой.

продолжение и уточнение следует