Лекция 12. Задачи по геометрии, математика.

Объекты: точки, векторы, отрезки, углы.

```
struct Point;
struct Vector;
struct Segment;
struct Line;
struct Ray;
struct Circle;
```

Можно составить библиотечку для элементарных базовых операций с такими объектами, чтобы проще записывать вычисления и построения в алгоритмах решения задач.

Некоторые задачи и алгоритмы вычислительной геометрии.

0. Угол между двумя векторами.

Какой из двух углов вычислять?

asin или atan2 ?

asin(x) — $[-\pi/2, \pi/2]$

acos(x) — $[0, \pi]$

atan(x) — $[-\pi/2, \pi/2]$

atan2(y,x) — $[-\pi, \pi]$

Для asin, acos проблема $|x| > 1$.

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.

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 из точки на ребра.

Как влияет выпуклость многоугольника?

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

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

— суммарный угол обхода сторон — устойчиво и $O(n)$.

5. Минимальный и максимальный диаметры (габариты).

Минимальная и максимальная ширина щели, в которую может “протиснуться” этот многоугольник.

Для максимального — расстояние между вершинами (max).

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

6. Покрытие отрезка на прямой набором заданных отрезков. Покрывает или нет?

Упорядочиваем по первой координате. Потом собираем связные объединения.

7. Расстояние между многоугольниками.

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

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

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

9. Клипирование отрезков по прямоугольнику.

Отсечение по x, потом отсечение по y.

10. Минимальный охватывающий круг.

Сжатие “охватывающего” кольца. С фиксацией на одной точке, на двух точках, на трех точках. Проверка на тупоугольный треугольник с переходом к наибольшей хорде.

11. Наибольший отрезок объединения.

Идейно совпадает с 6 задачей.

12. 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).

13. Выпуклая оболочка.

Алгоритм оборачивания по углу.

Алгоритм оборачивания сверху и снизу.

Алгоритм "разделяй и властвуй"

14. Раздувание или сжатие выпуклого многоугоьльника.

Наружу — просто сдвиг вершин по биссектриссам углов.

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

15. Минимальное покрывающее дерево.

Вычисляем все расстояния между точками. Упорядочиваем по возрастанию. Склеиваем в “кластеры” (связные компоненты). Регистрируем какая точка в какой кластер попала. Последовательно объединяем кластеры, исходя из минимального расстояния между ними.

16. Пересечение двух выпуклых многоугольников.

Проводим вертикальные линии через все вершины обоих многоугольников. Получается система “полос”, в каждой из которых может оказаться часть исходных многоугольников (трапеции). Теперь надо построить пересечения каждой пары таких трапеций и потом объединить их в общий многоугольник (удалить промежуточные вершины).

17. Генерация лабиринта.

Представим лабиринт в виде системы квадратных “клеток”, где каждая клетка может иметь от 0 до 3 стен из четырех возможных. В начальном состоянии все клетки имеют по 4 стены. Далее, стартуя из исходной клетки, удаляем стену, смежную с соседней замкнутой клеткой (пробиваем туда проход). Выбирая каждый раз случайное направление, строим некоторый “путь” до тех пор, пока это возможно (учитывая естественные ограничения). Если остались замкнутые клетки, берем любую из них и строим аналогично путь до выхода на уже ранее построенные пути. В результате между любыми двумя кдетками будет существовать путь, а оставшиеся стенки сформируют лабиринт.