

Векторы

Материал из Algocode wiki

Перейти к: навигация, поиск

Зафиксируем какую-нибудь прямоугольную декартову систему координат (обычно в олимпиадах других и не бывает). Начало координат назовём O , оси — O_x и O_y .

Вектором мы будем называть класс эквивалентности векторов на плоскости. Всякий вектор тогда задаётся парой из двух чисел (целых или действительных, в зависимости от задачи) — (x, y) . Тут же можно заметить, что каждой точке P на плоскости мы можем сопоставить её радиус-вектор OP , поэтому в коде для выражения точек и векторов нам будет достаточно одной структуры:

```
struct pt {  
    /* double */ int x{0}, y{0}; // выражение в фигурных скобках -- это значение по у  
    молчанию  
    pt (int x, int y) {  
        this->x = x;  
        this->y = y;  
    }  
};
```

Простейшие операции с векторами

Самое простое, что можно делать с векторами, это:

- Складывать/вычитать: получившийся вектор имеет своими координатами сумму/разность соответствующих координат векторов-аргументов.

```
// Здесь и везде далее, если не указано обратного, приводится код методов pt  
  
pt operator+(pt& other) {  
    return pt(x + other.x, y + other.y);  
};  
  
pt operator-(pt& other) {  
    return pt(x - other.x, y - other.y);  
};
```

- Получить вектор из одной точки в другую:

```
// это ещё один конструктор класса pt  
pt (pt& p1, pt& p2) {  
    return pt(p2.x - p1.x, p2.y - p1.y);  
}
```

Умножения векторов

Большой интерес из себя представляют операции умножения между векторами. Их нам понадобится 2:

- Скалярное произведение. Обозначается (v_1, v_2) , задаётся формулой $|v_1| \cdot |v_2| \cdot \cos \alpha$, где α — угол, на который надо повернуть v_1 , чтобы он стал коллинеарен v_2 . Геометрически равно произведению длины v_2 на длину проекции v_1 на v_2 .
- Косое (псевдоскалярное, векторное) произведение. Обозначается $[v_1, v_2]$, задаётся формулой $|v_1| \cdot |v_2| \cdot \sin \alpha$, где α — угол, на который надо повернуть v_1 , чтобы он стал коллинеарен v_2 . Геометрически равно площади параллелограмма, натянутого на v_1 и v_2 со знаком $+$, если поворот от v_1 к v_2 положительный и со знаком $-$ в противном случае.

(анти)Коммутативность произведений

1. $[v_1, v_2] = -[v_2, v_1]$
2. $(v_1, v_2) = (v_2, v_1)$

Связь с углом между векторами

- $(v_1, v_2) = 0 \iff$ угол между v_1 и $v_2 = \pm \frac{\pi}{2}$
- $(v_1, v_2) > 0 \iff$ угол между v_1 и v_2 — острый
- $(v_1, v_2) < 0 \iff$ угол между v_1 и v_2 — тупой
- $[v_1, v_2] = 0 \iff$ угол между v_1 и $v_2 \in \{0, \pi\}$
- $[v_1, v_2] > 0 \iff$ поворот от v_1 к v_2 в положительном направлении
- $[v_1, v_2] < 0 \iff$ поворот от v_1 к v_2 в отрицательном направлении

В дальнейшем, эти свойства произведений помогут нам в определении взаимного расположения точек и, например, прямой, поэтому важно эти свойства запомнить и понять.

Формулы для вычисления произведений

Геометрические формулы не дают простого ответа на то, как же посчитать произведение векторов, зная их координаты. Если посчитать длины векторов ещё можно с помощью теоремы Пифагора, то вот с определением угла между ними возникают трудности.

К счастью, для произведений есть довольно простые формулы:

- $(v_1, v_2) = x_1 \cdot x_2 + y_1 \cdot y_2$.
- $[v_1, v_2] = x_1 \cdot y_2 - x_2 \cdot y_1$.

```
//скалярное произведение
/*long double*/ long long operator*(pt& v1, pt& v2) {
    return v1.x * v2.x + v1.y * v2.y;
}

//векторное произведение
/*long double*/ long long operator%(pt& v1, pt& v2) {
    return v1.x * v2.y - v2.x * v1.y;
}

// Выбор операторов для умножений автор оставляет на усмотрение пишущего код.
// Сам автор использует именно те, которые приведены в примере выше.

// Отдельно автор хочет подчеркнуть, что использование операторов для умножения векторов
// является порочной практикой, поскольку вредит читаемости кода, а потому
// данная практика рекомендуется только в условиях олимпиад.
```

Доказательство этих формул следует из линейности обоих произведений (это надо доказать!) и

разложения по базисным векторам $\overline{(0, 1)}$ и $\overline{(1, 0)}$. Провести доказательство мы оставляем в качестве упражнения читателю.

Как найти точное значение угла между векторами

Предположим, мы умеем считать $\alpha = \operatorname{tg}^{-1}(\operatorname{tg}(\alpha))$. Величину в правой части равенства можно посчитать с помощью библиотечной функции *atan2* (есть в C++ и Python), подставив в неё в качестве аргументов векторное и скалярное произведения:

```
double angleBetween (pt& v1, pt& v2) {  
    return atan2(v1 % v2, v1 * v2);  
}
```

Поворот вектора

Поворот вектора на угол α задаётся следующим матричным тождеством:

$$\operatorname{Rot}_{\alpha}(\overline{(x, y)}) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(\alpha)x - \sin(\alpha)y \\ \sin(\alpha)x + \cos(\alpha)y \end{pmatrix}$$

В частности $\operatorname{Rot}_{90^{\circ}}(\overline{(x, y)}) = \overline{(-y, x)}$

Автор конспекта: Александр Гришутин

По всем вопросам пишите в telegram @rationalex (<https://t.me/rationalex>)

Источник — <https://wiki.algoricode.ru/index.php?title=Векторы&oldid=1654> (<https://wiki.algoricode.ru/index.php?title=Векторы&oldid=1654>)