Basic geometry cheatsheet

Brazilian Summer Camp 2018

Mikhail Tikhomirov

January 23, 2018

General advice

- Structure and encapsulate: objects = structs/classes, primitive actions = methods/functions
- Reduce & reuse code as much as possible
- Use integer computations as much as possible (take care for overflows!)
- Use `double` (or ever `long double`) instead of `float`
- Output as many decimal places as possible (`cout.precision(20)`, `cout << fixed`, `printf("%.20lf", x)`)

Dot-product

- For vectors a and b the *dot-product* $a \cdot b$ is defined as $r_1 r_2 \cos\alpha$, where r_1 and r_2 are lengths of a and b, α is the angle between a and b.
- $(x_1, y_1) \cdot (x_2, y_2) = x_1 x_2 + y_1 y_2$
- $(a + b) \cdot c = a \cdot c + b \cdot c$
- $a \cdot b$ is positive if $\alpha < 90°$, negative if $\alpha > 90°$, zero if $\alpha = 90°$
- $|a \cdot b|$ is equal to the length of a times the length of projection of b on a
- length of a is $\sqrt{a \cdot a}$

Cross-product

- The *dot-product* $a \times b$ is $r_1 r_2 \sin\alpha$, where r_1 and r_2 are lengths of a and b, α is the *directed* angle between a and b.
- $(x_1, y_1) \times (x_2, y_2) = x_1 y_2 - x_2 y_1$
- $(a + b) \times c = a \times c + b \times c$
- $a \times b$ is positive if $\alpha > 0$ (ccw shortest rotation from a to b), negative if $\alpha < 0$ (cw rotation), zero if $\alpha = 0$ or $180°$ (collinear)
- $|a \times b|$ is equal to the area of the parallelogram spanned by a and b

Lines

- Line equation: $ax + by + c = 0$ with $a \neq 0$ or $b \neq 0$
- (a, b) is a *normal vector* (orthogonal to the line), $(b, -a)$ is a *direction vector* (parallel to the line)
- Distance from a line to the point (x, y) is $|ax + by + c| / \sqrt{a^2 + b^2}$
- Cramer's rule for line intersection: $x = \dfrac{\begin{vmatrix} -c_1 & b_1 \\ -c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$, $y = \dfrac{\begin{vmatrix} a_1 & -c_1 \\ a_2 & -c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$

Convex hull (CH): Graham's algorithm

- Choose leftmost point p_0, sort all points in counter-clockwise order respective to p_0 (use comparator $(a - p_0) \times (b - p_0) > 0$)
- Construct CH incrementally in ccw order: when adding the current point p, erase last point of CH while the angle $\angle p_1 p_0 p > 0$, where p_1, p_0 are last two points of CH

Misc

- Rotate vector (x, y) by angle α: $(x \cos\alpha - y \sin\alpha, x \sin\alpha + y \cos\alpha)$
- Obtain α by $\cos\alpha$ and $\sin\alpha$: `atan2(r sin α, r cos α)` (more precise than `acos` and `asin`, less special cases, allows for arbitrary common multiplier r)
- Hypotenuse length: `hypot(a, b)` (more precise than `sqrt(a * a + b * b)`)
- Long double standard functions: `sinl(x)` instead of `sin(x)` for `long double x`