

АЛГОРИТМ 1 (АЛГОРИТМ МЭЛ'ОТ ОТСЕЧЕНИЯ МНОГОУГОЛЬНИКА ОТНОСИТЕЛЬНО ПРЯМОУГОЛЬНОЙ ОБЛАСТИ).

Вход: $P = \{(x_i, y_i) \mid 1 \leq i \leq n\}$ — набор вершин отсекаемого многоугольника в порядке их обхода по часовой стрелке, $(x_{\min}, y_{\min}), (x_{\max}, y_{\max})$ — координаты левого нижнего и правого верхнего углов окна отсеечения соответственно

Выход: n_1 — количество вершин в многоугольнике после отсеечения, $P_1 = \{(x'_i, y'_i) \mid 1 \leq i \leq n_1\}$ — набор вершин видимой части многоугольника в порядке их обхода по часовой стрелке

1. $k = 1$ (номер текущего ребра отсекаемого многоугольника),
 $n_1 = 0$ (количество ребер в результате отсеечения относительно текущей границы области видимости),
 $P_2 = \{\}$ (многоугольник — результат отсеечения относительно текущей границы области видимости).
2. $(x_{start}, y_{start}) = P[n]$, $C_{start} = ExtCode(x_{start}, y_{start})$.
3. Если $k > n$, закончить алгоритм, иначе — переход к шагу 4;
4. $(x_{end}, y_{end}) = P[k]$, $C_{end} = ExtCode(x_{end}, y_{end})$.
5. (Начало алгоритма Козна—Сазерленда) $C_1 = C_{start}$, $C_2 = C_{end}$, $(x_1, x_2) = (x_{start}, y_{start})$, $(x_2, y_2) = (x_{end}, y_{end})$.
6. $clip = \mathbf{false}$ (отметка о том, что начальная точка была отсечена),
 $flip = \mathbf{false}$ (отметка о том, что начальная и конечная точки переставлены местами).
7. Если $(C_1|C_2)\&15 = 0$ — отрезок полностью видим: переход к шагу 8, а иначе переход к шагу 10.
8. Если $flip$, то поменяем местами значения x_1 с x_2 , y_1 с y_2
9. $segm = \mathbf{true}$ (отметка о том, что часть отрезка попала в область видимости) и переход к шагу 14.
10. Если $C_1\&C_2\&15 \neq 0$, то отрезок полностью невидим: $segm = \mathbf{false}$ и переход к шагу 14.
11. (Отрезок может быть частично видим) Если $C_1\&15 = 0$ — поменяем местами значения x_1 с x_2 , y_1 с y_2 , C_1 с C_2 , $flip = !flip$.
12. Найдем точку пересечения отрезка с одной из прямых, ограничивающих область видимости и перенесем туда точку (x_1, y_1) :
 - (а) Если $C_1\&15 \neq 0$ и $!flip$, то $clip = \mathbf{true}$;

(b) Если $C_1 \& 1 \neq 0$, то

$$\begin{aligned} y_1 &= y_2 - (x_2 - x_{\min}) \frac{y_2 - y_1}{x_2 - x_1} \\ x_1 &= x_{\min} \end{aligned}$$

(c) Если $C_1 \& 2 \neq 0$, то

$$\begin{aligned} y_1 &= y_2 - (x_2 - x_{\max}) \frac{y_2 - y_1}{x_2 - x_1} \\ x_1 &= x_{\max} \end{aligned}$$

(d) Если $C_1 \& 4 \neq 0$, то

$$\begin{aligned} x_1 &= x_2 - (y_2 - y_{\min}) \frac{x_2 - x_1}{y_2 - y_1} \\ y_1 &= y_{\min} \end{aligned}$$

(e) Если $C_1 \& 8 \neq 0$, то

$$\begin{aligned} x_1 &= x_2 - (y_2 - y_{\max}) \frac{x_2 - x_1}{y_2 - y_1} \\ y_1 &= y_{\max} \end{aligned}$$

(f) $C_1 = ExtCode(x_1, y_1)$

13. Перейти к шагу 7.

14. Алгоритм отсечения отрезка Коэна—Сазерленда окончен. Перейти к продолжению алгоритма отсечения многоугольника — к шагу 15.

15. $C_2 = C_{end}$,

16. Если $segm$,

(a) Если $clip$, (добавление точки пересечения) $n_1 = n_1 + 1$, $P_1[n_1] = (x_1, y_1)$.

(b) (добавление конечной точки ребра) $n_1 = n_1 + 1$, $P_1[n_1] = (x_2, y_2)$.

(c) Переход к шагу 18.

17. Иначе (если ребро полностью невидимо, рассматриваем особые случаи: проверяем, нужно ли добавлять угловые точки области видимости в многоугольник, и если нужно, то определяем какие)

(a) Если $C_{end} \& 16 \neq 0$ (В этом случае может быть добавлено 2 угловых точки),

- Если $C_{start} \& C_{end} \& 15 = 0$,

i. Если $C_{start} \& 16 = 0$,

$$C_1 = C_{end} + Tcc(C_{start});$$

иначе

A. $(x_1, x_2) = (x_{start}, y_{start})$, $(x_2, y_2) = (x_{end}, y_{end})$, $C_{11} = C_{start}$, $C_{12} = C_{end}$;

B.

$$x_{mid} = \frac{x_1 + x_2}{2},$$

$$y_{mid} = \frac{y_1 + y_2}{2};$$

C. $C_1 = ExtCode(x_{mid}, y_{mid})$;

D. Если $C_1 \& 16 \neq 0$ (средняя точка попала в угловую область)

– Если $C_1 = C_{12}$, то $(x_2, y_2) = (x_{mid}, y_{mid})$, $C_{12} = C_1$ и переход к шагу 17(a)B;

– Если $C_1 = C_{11}$, то $(x_1, y_1) = (x_{mid}, y_{mid})$, $C_{11} = C_1$ и переход к шагу 17(a)B;

иначе (средняя точка попала в неугловую область)

– Если $C_1 \& C_{12} \neq 0$,

$$C_1 = C_{11} + Tcc(C_1)$$

иначе

$$C_1 = C_{12} + Tcc(C_1)$$

ii. (Добавление 1-й угловой точки) $n_1 = n_1 + 1$, $P_1[n_1] = Angle(C_1)$;

• Переход к шагу 18.

(b) Иначе

• Если $C_{start} \& C_{end} \& 15 = 0$,

– Если $C_{start} \& 16 \neq 0$,

$$C_2 = C_{start} + Tcc(C_{end}),$$

иначе

$$C_2 = C_{start} + C_{end} + 16;$$

18. (Тест на добавление угловой точки) Если $C_2 \& 16 \neq 0$, то $n_1 = n_1 + 1$ и $P_1[n_1] = Angle(C_2)$;

19. $k = k + 1$, $C_{start} = C_{end}$, $(x_{start}, y_{start}) = (x_{end}, y_{end})$ и переход к шагу 3.

АЛГОРИТМ 2 (EXTCODE: ВЫЧИСЛЕНИЕ РАСШИРЕННОГО КОДА ОБЛАСТИ ДЛЯ ТОЧКИ).

Вход: (x, y) — координаты точки, (x_{\min}, y_{\min}) , (x_{\max}, y_{\max}) — координаты левого нижнего и правого верхнего углов окна отсечения соответственно

Выход: C — код области, в которую попадает заданная точка

1. $C = 0$;
2. Если $x < x_{\min}$, то C увеличить на 1;
 - (a) Если $y < y_{\min}$, то C увеличить на 20;
 - (b) Если $y > y_{\max}$, то C увеличить на 24;
3. Если $x > x_{\max}$, то C увеличить на 2;
 - (a) Если $y < y_{\min}$, то C увеличить на 20;
 - (b) Если $y > y_{\max}$, то C увеличить на 24;
4. Если $y < y_{\min}$, то C увеличить на 4;
5. Если $y > y_{\max}$, то C увеличить на 8;

АЛГОРИТМ 3 (ANGLE: КООРДИНАТЫ УГЛА ОБЛАСТИ ВИДИМОСТИ ПО КОДУ УГЛОВОЙ ОБЛАСТИ).

Вход: C — код угловой области

Выход: (x, y) — координаты соответствующей угловой точки области видимости

1. Если $C \& 1$, $x = x_{\min}$;
2. Если $C \& 2$, $x = x_{\max}$;
3. Если $C \& 4$, $y = y_{\min}$;
4. Если $C \& 8$, $y = y_{\max}$;
5. выдать (x, y) ;

АЛГОРИТМ 4 (ТСС: КОРРЕКЦИЯ КОДА ОБЛАСТИ ПРИ НЕОБХОДИМОСТИ ДОБАВЛЕНИЯ УГЛОВОЙ ТОЧКИ).

Вход: C — код неугловой области

Выход: $corr$ — коррекция кода

1. Если $C \& 1$, $corr = -1$;

2. Если $C \& 2$, $corr = 1$;
3. Если $C \& 4$, $corr = -4$;
4. Если $C \& 8$, $corr = 4$;
5. выдать $corr$;