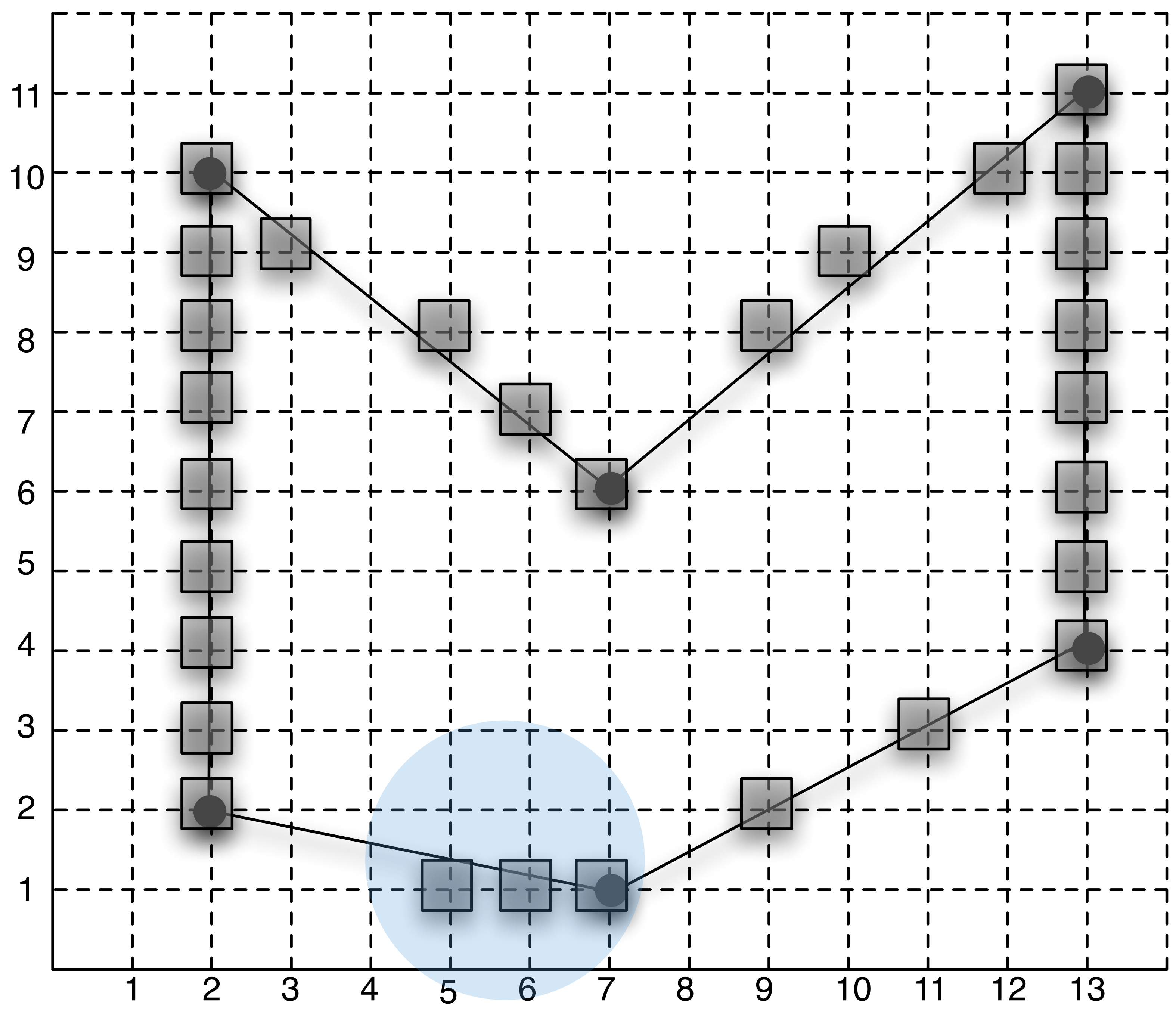
ALGORITMOS DE RELLENADO DE POLÍGONOS

Computación Gráfica

Recorrido de aristas

- ¿Cómo encontrar los extremos sobre cada horizontal?
    - rellenar cada arista con el algoritmo del punto medio.



- Algunos pixels están al exterior del polígono.
- Problema si dibujamos varios polígonos diferentes con diferentes colores.
- Podría mostrar polígonos disjuntos como juntos.
- No vamos a utilizar el algoritmo del punto medio.
- Algoritmo incremental: elegir los pixels a la derecha de la arista ideal si se trata de una arista izquierda y viceversa.
- Memorizamos los pixels extremo generados por este algoritmo en una tabla.

Casos especiales

- Intersección de aristas con $\mathbb{Z}^2$:
  - si el pixel está en una arista izquierda del polígono, dibujarla. Si no, no dibujarla.
  - si un pixel se encuentra en el extremo de dos aristas del polígono, no contar ambas aristas para la regla de paridad.
- Aristas horizontales
  - dibujar las aristas inferiores pero no las superiores.

Algoritmo para recorrer las aristas de un polígono

- Primera parte del algoritmo: encontrar las intersecciones de cada línea horizontal con las aristas del polígono.
- Cada arista del polígono va de un pixel (x_{bajo}, y_{bajo}) a un pixel (x_{alto},y_{alto}),
- Variar la coordenada y de y_{bajo} a y_{alto} obteniendo la relación siguiente:

$$y_{i+1} = y_i + 1$$
$$x_{i+1} = x_i + \frac{1}{m}$$

$$\frac{1}{m} = \frac{x_{alto} - x_{bajo}}{y_{alto} - y_{bajo}} = \frac{a}{b}$$

- Considerar una variable entera *incremento*, tal que *incremento/b* represente la parte fraccionaria de x_i.
- La variable *incremento* será aumentada por el valor de a en cada iteración pero dibujamos un pixel cuya abcisa es un valor redondeado de x_i.

Algoritmo para recorrer las aristas de un polígono

- El valor redondeado será incrementado hasta que la cantidad *incremento/b* rebase *1*.
- Si nombramos *a* la parte entera de *incremento/b*, entonces la abscisa del pixel a dibujar es aumentada con el valor de *a*.
- La variable *incremento* debe permanecer entre *0* y *b-1* restando a incremento un cierto número de veces el valor de *b*: esto es, *a x b*.
- La inicialización de la variable *incremento* depende de la pendiente de la recta y si la arista es derecha o izquierda del polígono.

Algoritmo para recorrer las aristas de un polígono

- Para una arista izquierda con $a>0$ (pendiente positiva):

```
int increment = b-1;
```

- Para una arista derecha con $a>0$ (pendiente positiva):

```
int increment = -1;
```

- Para una arista izquierda con $a\leq 0$ (pendiente negativa):

```
int increment = 0;
```

- Para una arista derecha con $a\leq 0$ (pendiente negativa):

```
int increment = -b;
```

Arista izquierda con pendiente positiva

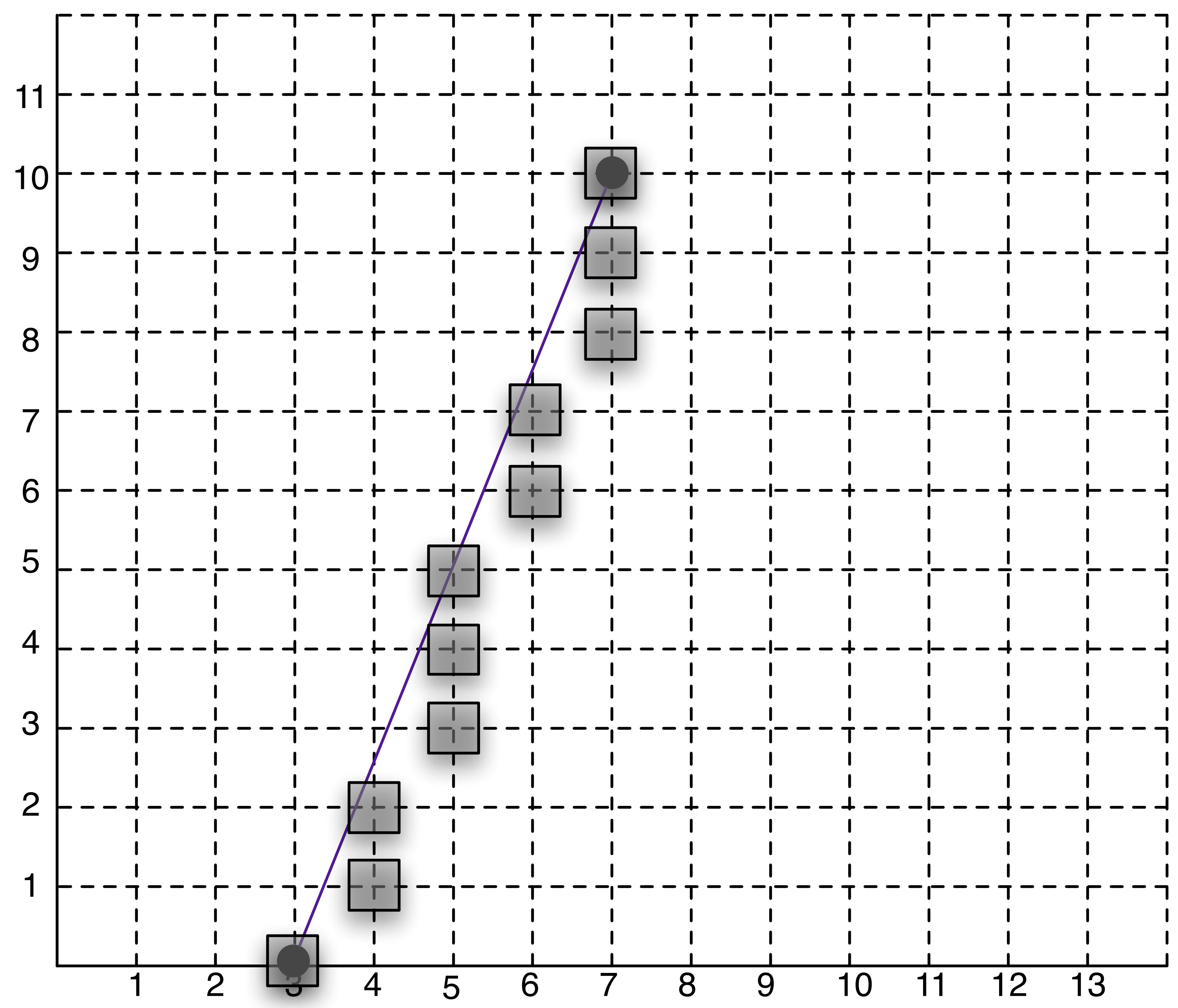
```
void izq_pend_pos( int xbajo, int ybajo, int xalto, int yalto, unsigned char color )
{
   int y;
   int x = xbajo;
   int a = xalto - xbajo;
   int b = yalto - ybajo;
   int increment = b-1;
   int Q;

   for( y=ybajo; y <= yalto; y++ )
   {
    PutPixel( x, y, color );
    increment += a;
    Q = increment/b;
    x += Q;
    increment -= Q * b;
   }
}
```

$(x_{bajo}, y_{bajo}) = (3,0)$ $(x_{alto}, y_{alto}) = (7,10)$

$a = x_{alto} - x_{bajo} = 7\text{-}3 = 4$

$b = y_{alto} - y_{bajo} = 10\text{-}0 = 10$

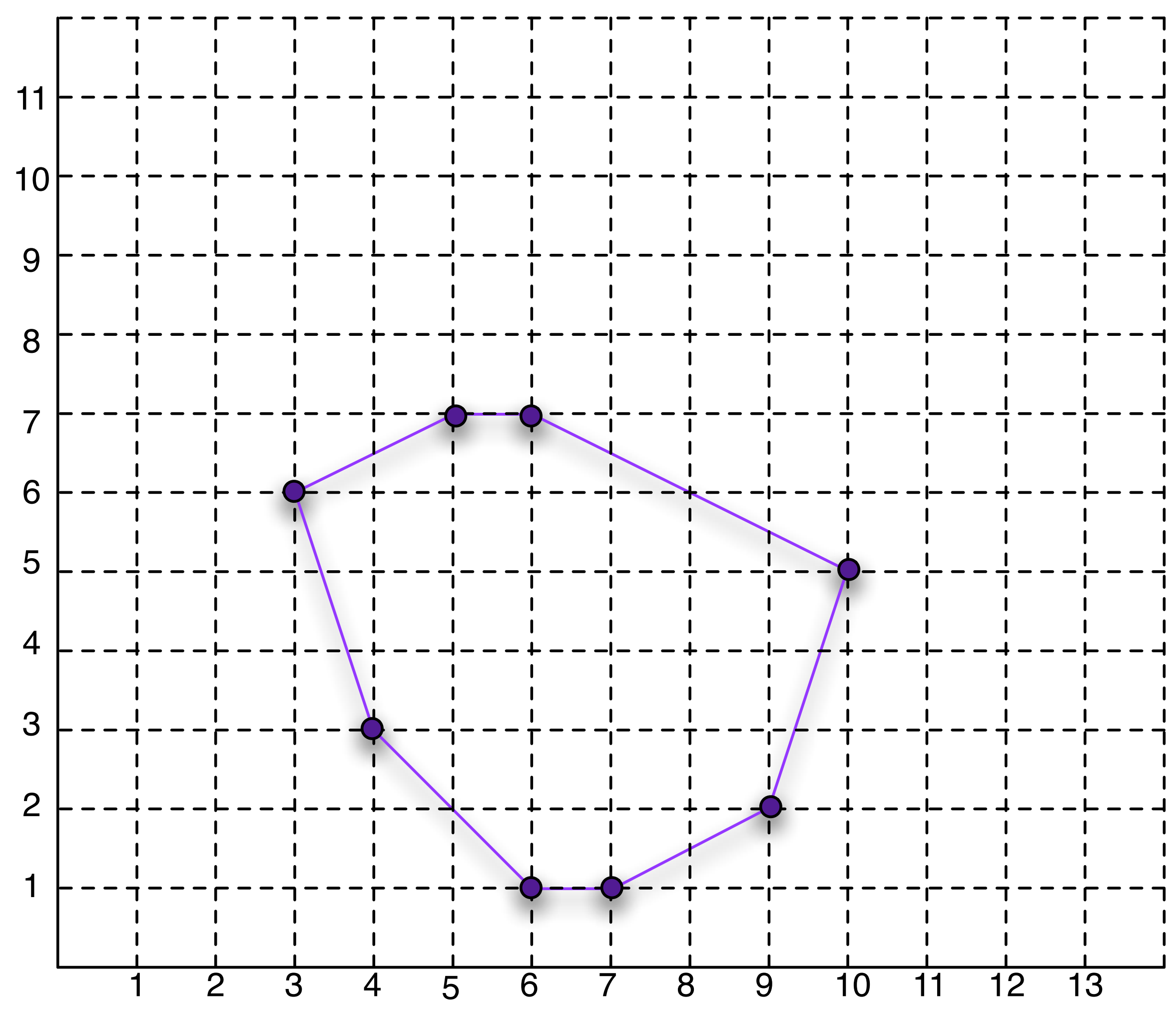
```
void izq_pend_pos( int xbajo, int ybajo, int xalto,
int yalto, unsigned char color )
{
    int y;
    int x = xbajo;
    int a = xalto – xbajo;
    int b = yalto – ybajo;
    int increment = b-1;
    int Q;

    for( y=ybajo; y <= yalto; y++ )
    {
     PutPixel( x, y, color );
     increment += a;
     Q = increment/b;
     x += Q;
     increment -= Q * b;
    }
}
```



y	0	1	2	3	4	5	6	7
increment	9	13 > 10 → 3	7	11 > 10 → 1	5	9	13 > 10 → 3	7
increment/b	9/10	13/10 → 3/10	7/10	11/10 → 1/10	5/10	9/10	13/10 → 3/10	7/10
Pixel	(3,0)	(4,1)	(4,2)	(5,3)	(5,4)	(5,5)	(6,6)	(6,7)

Polígonos Convexos

- Para cada horizontal, tenemos a lo más un segmento incluido en el polígono.
- Ir de abajo hacia arriba en el polígono recorriendo simultáneamente dos aristas, la izquierda y la derecha con el algoritmo antes visto.
- Mantener una tabla de aristas activas TAACONV: para un valor dado de y, incluir las dos aristas del polígono que intersecan la horizontal de altura y.



$y = 4$

6	4	-1	3	5	9	1	3
y_{alto}	x	a	b	y_{alto}	x	a	b

$y = 5$

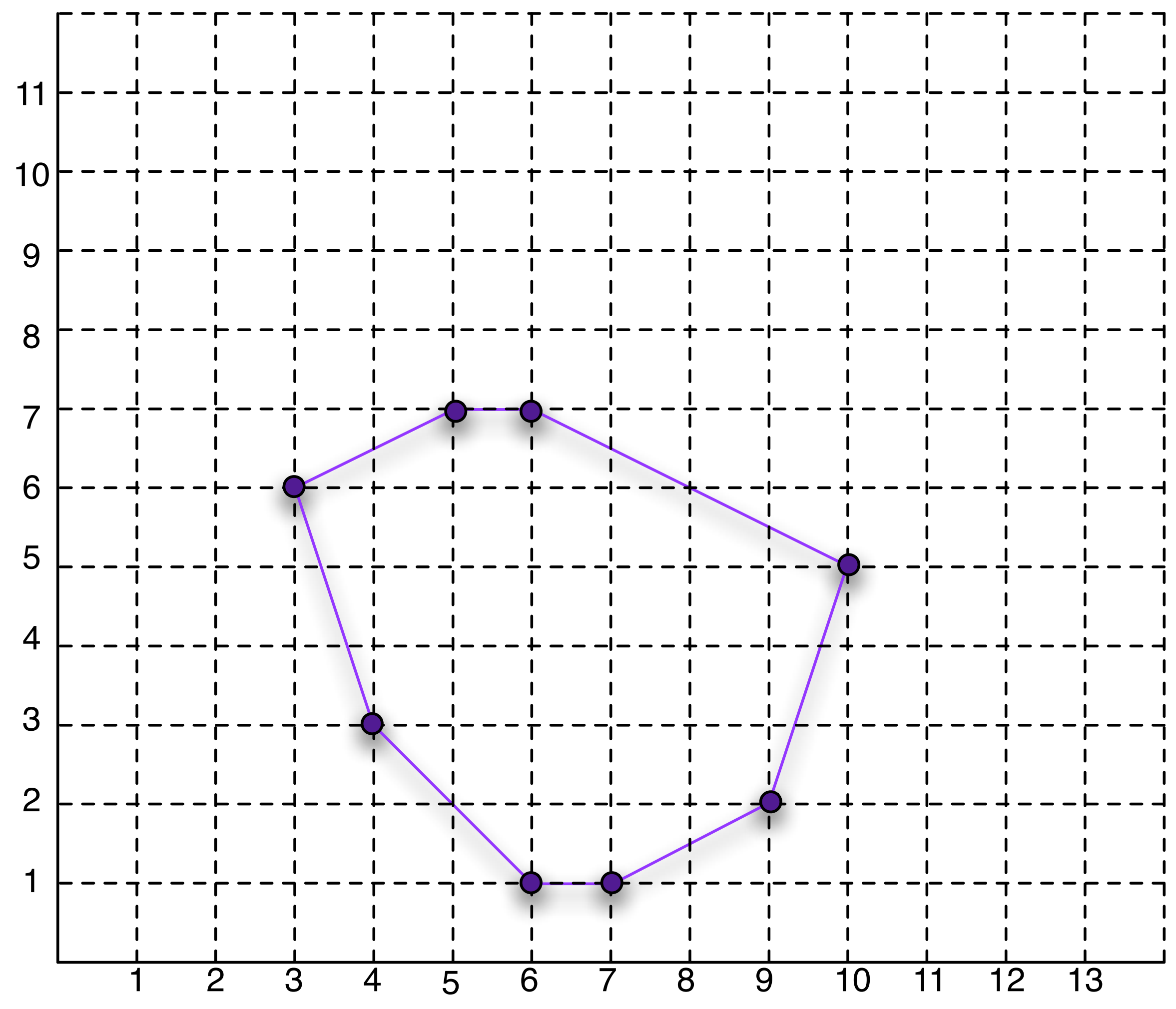
6	4	-1	3	7	9	-4	2
y_{alto}	x	a	b	y_{alto}	x	a	b

$y = 6$

7	3	2	1	7	7	-4	2
y_{alto}	x	a	b	y_{alto}	x	a	b

Polígonos Convexos

- Para incluir rapidamente una nueva arista, tal que $y = y_{bajo}$, construimos una tabla global de aristas (TACONV) concatenando todas las aristas no-horizontales del polígono, clasificadas por su y_{bajo}.

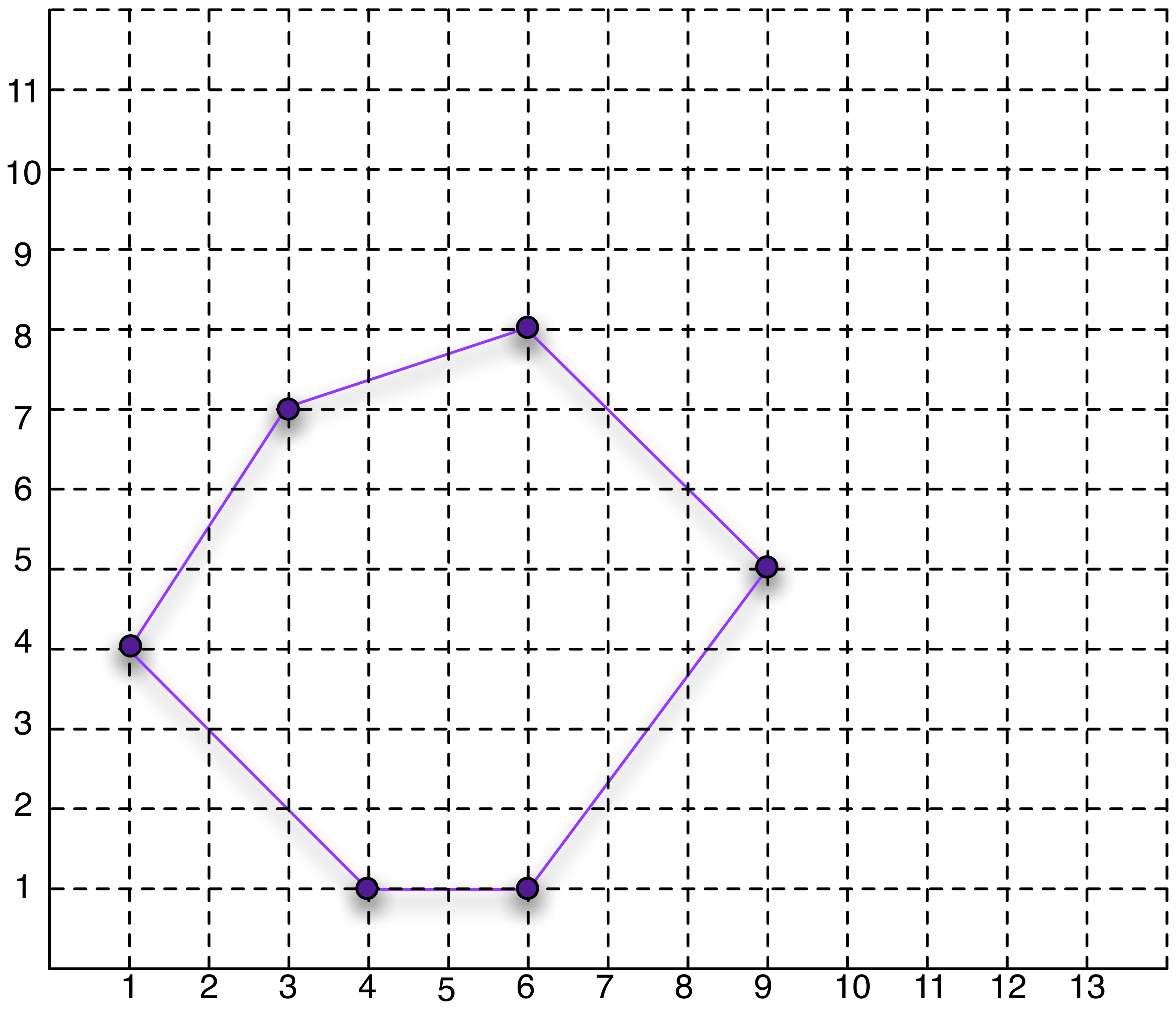


	y_{alto}	x_{bajo}	a	b	y_{alto}	x_{bajo}	a	b
7 →	7	3	2	1				
6 →	7	10	-4	2				
5 →								
4 →	6	4	-1	3				
3 →	5	9	1	3				
2 →	3	6	-2	2	2	7	2	1

Polígonos Convexos

- Inicializar y con el valor mínimo de y_{bajo}: $miny_{bajo}$.
- Copiar las dos aristas de la tabla TACONV a la altura 0 (la más baja corresponde a y) en la tabla de aristas activas TAACONV. Inicializar las dos variables `increment`.
- Repetir mientras que y sea estríctamente inferior al valor más grande posible de y_{alto}.
  - Dibujar los pixels entre los dos pixels (x,y) correspondientes a los dos elementos de TAACONV.
  - Remplazar las aristas de TAACONV que tienen su y_{alto} igual a y por las aristas de la tabla TACONV para aquellas que $y_{bajo}=y$ (que se encuentran a la altura y-$miny_{bajo}$. Para estas aristas inicializar las variables `increment`.
  - Actualizar los valores de x en la tabla TAACONV para incrementar y, siguiendo el algoritmo de recorrido de aristas.
  - Incrementar y.

Ejemplo



	y_{alto}	x_{bajo}	a	b	y_{alto}	x_{bajo}	a	b
8	8	3	3	1				
7								
6	8	9	-3	3				
5	7	1	2	3				
4								
3								
2	4	4	-3	3	5	6	3	4

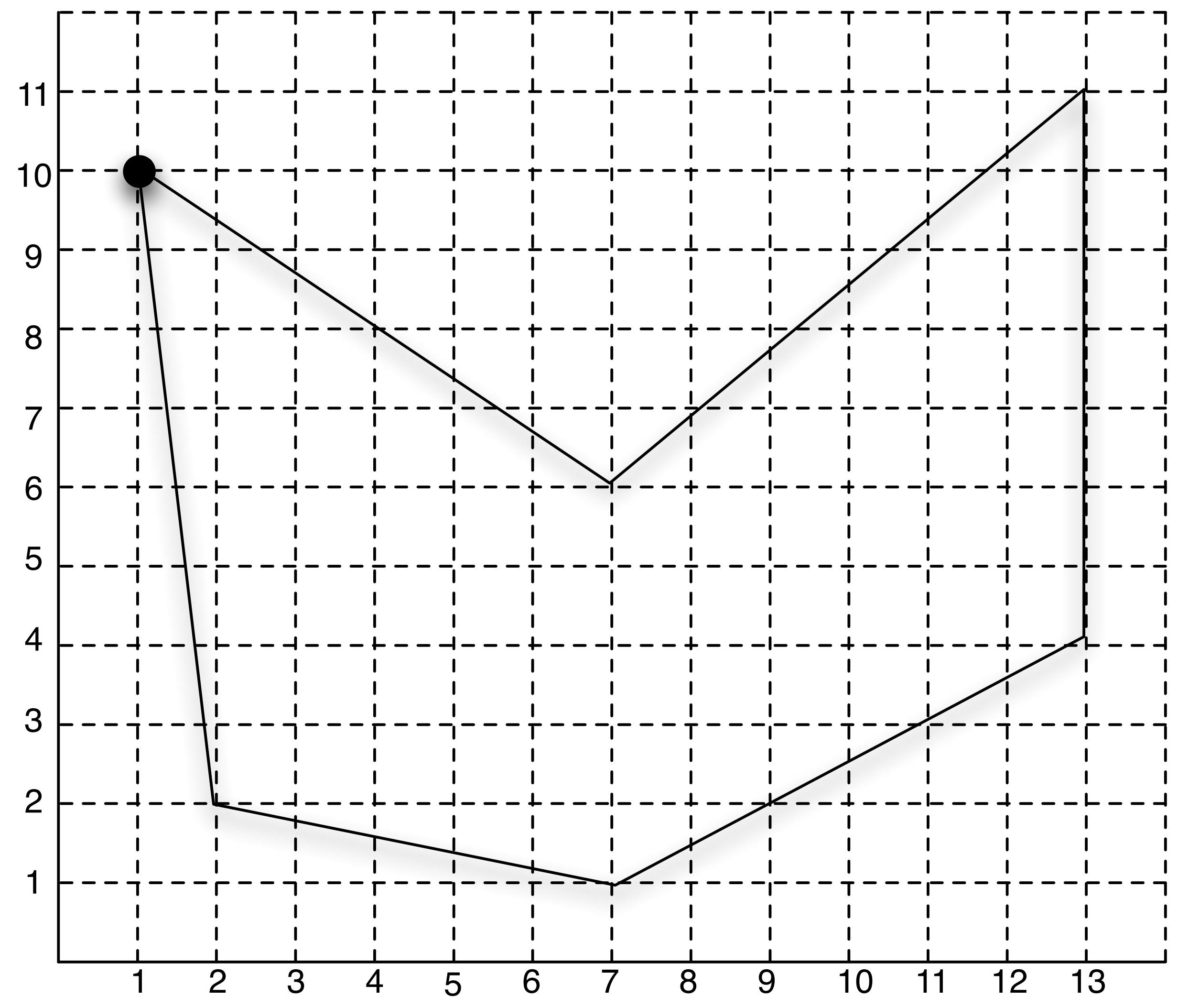
TAACONV →

4	4	-3	3	5	6	3	4

```
increment[0] = 0;    // arista izquierda con pendiente negativa
increment[1] = -1;   // arista derecha con pendiente positiva
```

Polígonos Arbitrarios

- Mismo principio que los polígonos convexos.
  - el número de aristas del polígono que intersecan una horizontal dada no es igual a 2.
  - la tabla de aristas activas (TAA) no es una tabla con dos estructuras sino una lista encadenada.
  - la variable incremento debe multiplicarse por el número de aristas en la lista TAA.
  - la tabla global de aristas (TA) es una tabla de listas encadenadas.



TAA para $y=5$

10	2	-1	8	-3		→	11	13	0	7	0	
y_{alto}	x	a	b	incremento								

TAA para $y=6$

10	2	-1	8	-4		→	10	7	-6	4	-4		→	11	7	6	5	4		→	11	13	0	7	0	

TAA para $y=7$

10	2	-1	8	-5		→	10	5	-6	4	-2		→	11	9	6	5	0		→	11	13	0	7	0	