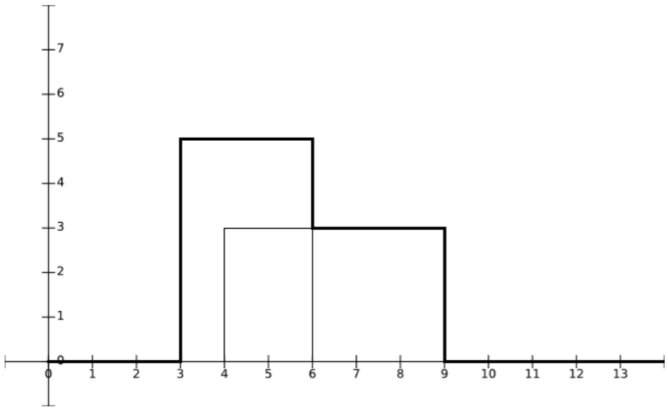
# Programación Funcional V

1) **Dibujando la silueta de los edificios mediante divide y vencerás (7 puntos)**

A partir de una lista de edificios rectangulares dados por ternas de números enteros (a,b,h), donde a es la primera coordenada x del edificio, b la segunda coordenada x del edificio, y h la altura del edificio. Por ejemplo, (3,6,5) y (4,9,3) representan dos edificios, el primero se extiende desde la coordenada x 3 hasta la coordenada x 6 con una altura de 5, y el segundo desde la coordenada x 4 hasta la 9, con una altura de 3. La silueta de estos edificios empieza en la coordenada 3, donde vale 5, cambia en la coordenada 6 donde vale 3 y vuelve a cambiar en la coordenada 9, con valor 0. Esta silueta se puede representar mediante una lista de pares (x,h), donde x es la posición en la que cambia la silueta y h la altura a la que cambia en esa posición. En el ejemplo, [(3,5),(6,3),(9,0)] .




Buena explicación en inglés:
https://www.youtube.com/watch?v=GSBLe8cKu0s
https://briangordon.github.io/2014/08/the-skyline-problem.html

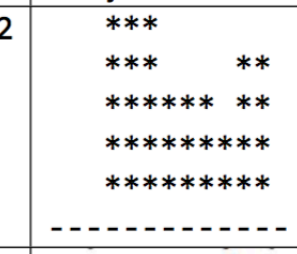
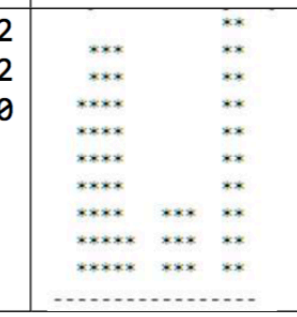
Nuestro objetivo será identificar en que posiciones x cambia la altura de los edificios. La altura de la silueta sólo puede cambiar en las posiciones x donde comienza o finaliza un edificio.

La implementación está detallada en el fichero .hs adjunto. Deben implementarse las funciones explicadas en el fichero. Si se necesitan implementar funciones auxiliares, éstas se implementarán como subfunciones utilizando la cláusula where.

2) **Imprimiendo en pantalla la silueta de los edificios mediante asteriscos.**

En la siguiente tabla se pueden ver dos ejemplos de visualización en pantalla. El algoritmo

a implementar viene descrito en el fichero .hs

| HASKELL | | |
|---|---|---|
| Lista de edificio | skyline | dibujo |
| [(3,6,5),(4,9,3),(8,11,2),(10,12,4)] | [(3,5),(6,3),(9,2),(10,4),(12,0)] |  |
| [(2,5,7),(3,6,9),(5,7,2),(9,12,3),(14,16,10)] | [(2,7),(3,9),(6,2),(7,0),(9,3),(12,0),(14,10),(16,0)] |  |