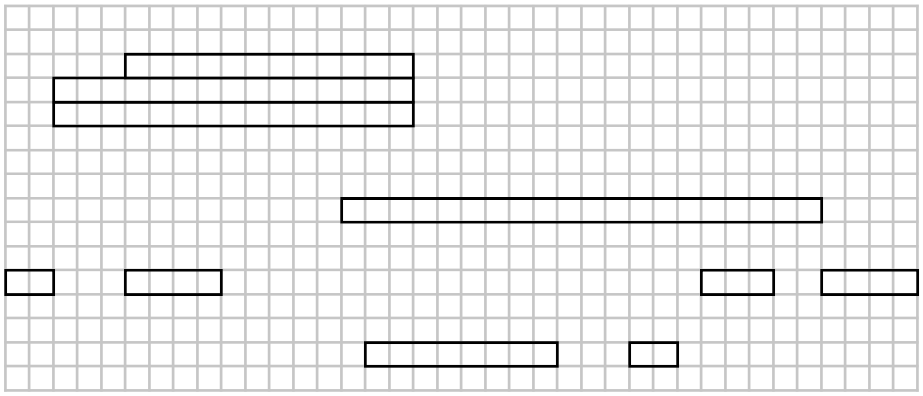
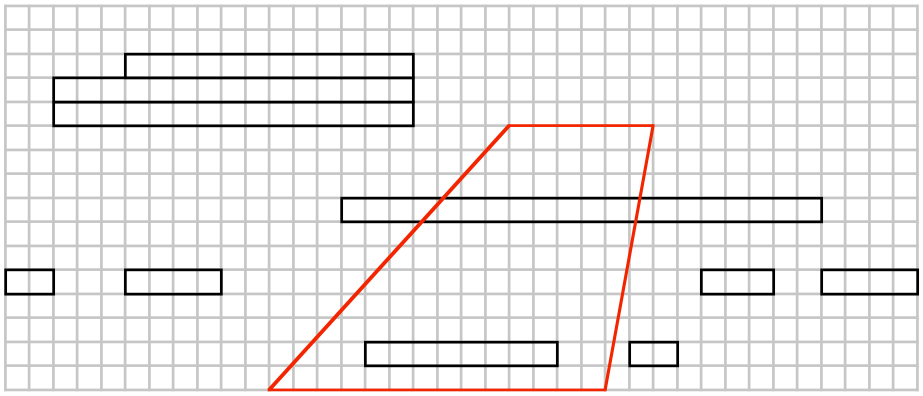
Raster Segments

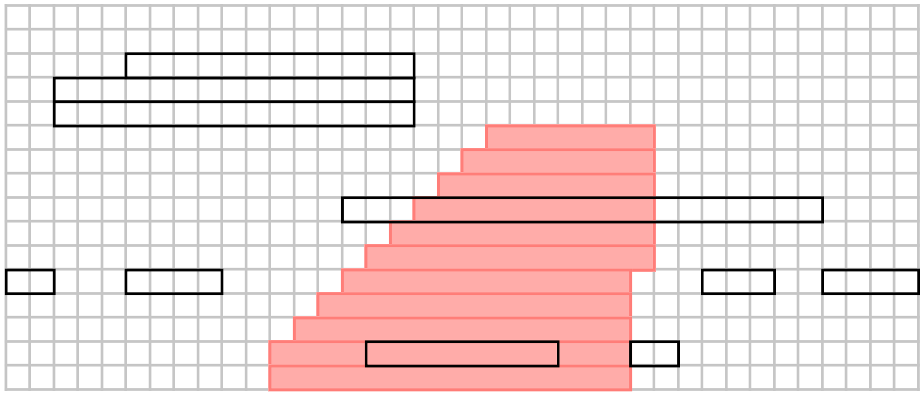
(See the struct definition RasterRow_ in the file “definitions.h”.)



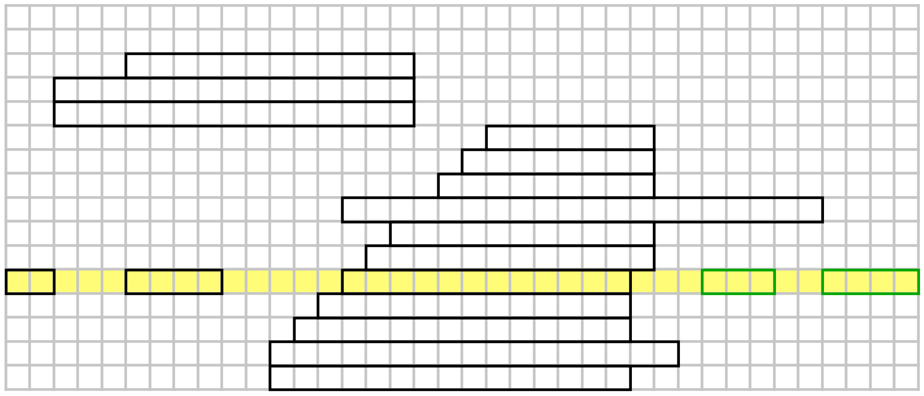
This is a pixel grid. The black rectangles are horizontal segments of continguous pixels that need to be re-rendered, as stored in the “Raster” array. Notice that a pixel row may have more than one segment. The maximum number of segments per row is defined as MAX_RENDER_SEGS_PER_ROW — in this example that is assumed to have a value of 4.



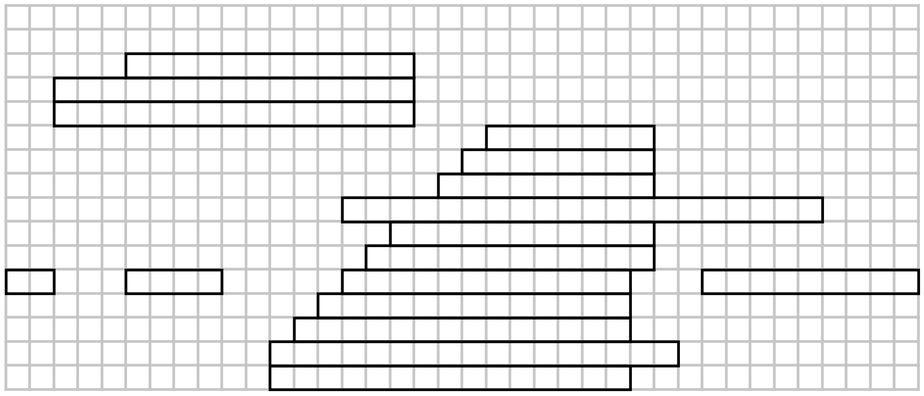
The app decides that the parallelagram (red) has been invalidated and needs to be included in the pixels-to-be-rendered (Raster) array.



To achieve that, these new (pink) segments need to be added to the array. So they are, and are *merged* into any pre-existing segments which they overlap or touch.



Notice that the result of the segment-merging process will give one row (yellow) a total of 5 segments, which exceeds the array's maximum of 4 segments per row. To solve this, the code will join the closest pair of segments (green).



Done! Because of the joining of the green segments, two extra pixels were included that actually didn't need to be re-rendered. That is simply unavoidable when the array limit (4) is reached.