Programming Assignment on Special Case of Skylines Problem

Nitin Singh Patel

Explanation

The `merge` function in the provided code contains several `if-else` conditions to handle different scenarios encountered during the merging process of two sub-half outlines. These conditions are crucial for determining the correct order of lines in the merged vector and final set of visible outlines.

1. Firstly, there are conditions to check whether two lines **intersect** or not. If they do intersect, the code calculates the intersection point and breaks the lines into segments accordingly, ensuring that the merged vector maintains the correct order.
2. The code also checks for coincident lines using `fabs` function to handle cases where lines overlap perfectly. If the lines coincide, the code appropriately updates the indices and continues the iteration.
3. Additionally, the `if-else` conditions handle cases where lines do not intersect. Depending on the relative positions of the lines, such as **disjoint, contained, or partially contained**, the code determines how to merge them properly. For example, if one line is completely contained within the other, the containing line may need to be split into segments.
4. Within these conditions, there are further checks based on the starting x-coordinate of the lines. This helps in determining the correct order when merging the lines of sub-halfs. Specifically, if one line starts before the other, it should be processed first to maintain the sorted order.
5. Moreover, the code employs numerical tolerance checks using the `1e-9, 1e-6, etc.` threshold to handle floating-point arithmetic errors, ensuring accurate comparisons and calculations when dealing with line slopes and intersections.

Basically, while merging outlines of different halfs, we start iterations segment by segment from each of left and right half. Then, we need to analyse the possible configurations for two line segments which are as follows:

- Both lines intersect
- One line's start and end x-coordinates are entirely contained in range of another line's start and end x-coordinates
- One line's start x-coordinate is between another line's start and end x-coordinates

We need to elaborate these configurations, and make different cases based on these which are pretty well handled in code and briefly discussed above.

Thank You for Reading Explanation