1. When clipping a point against a line, under what condition is the point kept? What does this depend on?
2. When clipping a segment against a line (half-plane), what cases do we distinguish based on the positions of the endpoints relative to the clipping line? What happens if one endpoint of the segment lies on the clipping line?
3. Calculation Exercise: let $\mathbf{p} = (4, 2)$ and $\mathbf{q} = (9, -5)$. What are the endpoints $\mathbf{p}'$ and $\mathbf{q}'$ of the segment after clipping against the line $y = -1$ $(= y_{\text{max}})$, i.e., removing the points lying "above" the line? After that, also clip against the line $y = -4$ $(= y_{\text{min}})$, which removes the points lying "below" the line; what are $\mathbf{p}''$ and $\mathbf{q}''$?
4. In the Cohen–Sutherland clipping algorithm, in what case can we discard the entire segment with certainty?
5. In the Cohen–Sutherland clipping algorithm, the codes of the two endpoints of a segment are $(1000, 0101)$. Is it possible to discard the entire segment? Is it possible to keep part of the segment? Is either outcome certain (discarding the whole segment or keeping part of it)?
6. Calculation Exercise: Among the following pairs of points with Cohen–Sutherland codes, which segments are guaranteed to lie outside the clipping window? $(0101, 0110)$, $(1001, 0010)$, $(0100, 0001)$, $(1000, 0000)$, $(0000, 0000)$
7. Outline how the Sutherland–Hodgman polygon clipping algorithm works.
8. Calculation Exercise: Given a triangle in the plane with vertices $(4, 4)$, $(14, 14)$, and $(-6, 24)$. Clip it sequentially against the lines $x = 0$, $y = 0$, $x = 10$, and $y = 10$, keeping the points in the half-planes defined by the corresponding normals $\mathbf{n} = (1, 0)$, $\mathbf{n} = (0, 1)$, $\mathbf{n} = (-1, 0)$, and $\mathbf{n} = (0, -1)$. What will be the coordinates of the vertices of the resulting polygon after clipping? (Sutherland–Hodgman)
9. How can we determine on which side of a plane a point lies in 3D?
10. In the Bresenham line-drawing algorithm, is it guaranteed that we always "hit" the start and end points of the line segment?
11. In the Bresenham algorithm, is it necessary to use floating point values?
12. The first version of the Bresenham algorithm we discussed (the one we already eliminated floating point operations from) only worked for a certain set of lines. Which lines were these? How did we make it work for all lines while still using essentially the same algorithmic core?
13. Calculation Exercise: Consider a triangle with vertices $(1, 1)$, $(9, 5)$, and $(5, 9)$, listed in counterclockwise order. What are the direction vectors of its edges, and what are the normals derived from them? How can we decide for a pixel p whether it lies inside the triangle or not? (Reminder: the $+90°$ rotation of (x, y) is $(-y, x)$, and the $-90°$ rotation is $(y, -x)$.)
14. How can a triangle be rasterized with the help of barycentric coordinates?