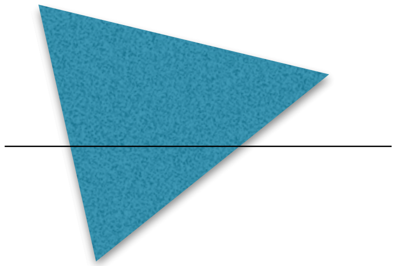
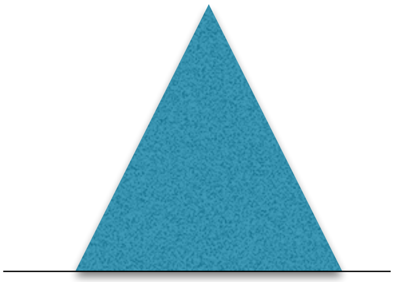
Q1: What are the three different configuration cases when determining the intersection of a pixel row with a triangle edge? In all three cases, what simple criterion can one use to determine whether the triangle edge overlaps the pixel row at all?



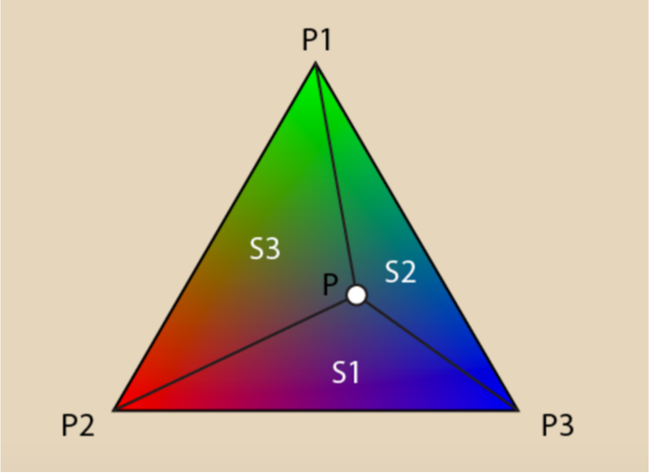
A1:
Case 1 - the intersection point is on the triangle edge, e.c. left edge and right edge in the triangle above. We can use y coordinate value to distinguish this case, where y coordinate value of intersection point is between y value of edge points.

Case 2 - the intersection point is on the extend of triangle edge, e.c. top edge in the triangle above. In this case, both y value of edge points are either greater than y value of pixel row, or smaller than pixel row.



Case 3 - triangle edge is collinear with pixel row, see picture above. In this case, edge and pixel row have 0/2 intersection points, which can be used as a judging condition for this case.

Q2: How might one use barycentric interpolation to determine whether or not a given point in space lies within the bounds of a triangle? In rasterization, would this method be more efficient than row bound checking for determining which pixels lie within a triangle? Why or why not?



A2: We need to calculate the area of triangle P1P2P3, noted as S, and areas of PP2P3(S1), PP1P3(S2), PP1P2(S3), if S1+S2+S3 = S, then point P lies within the bounds of a triangle, otherwise it's not.

It will be more efficient than row bound check. For area calculation, given coordinates of three corners be (x1, y1), (x2, y2) and (x3, y3), then area can be computed as [x1(y2 – y3) + x2(y3 – y1) + x3(y1-y2)]/2, it clearly requires less computation than row bound checking.