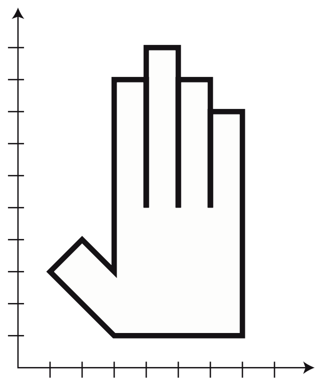
CMSC 435/634 Problem Set 1

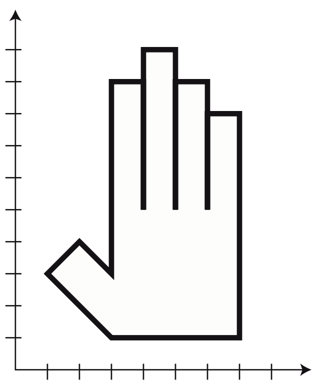
This problem set is provided for study purposes only, it will not be graded. A solution will be provided in one week.

BSP Trees:

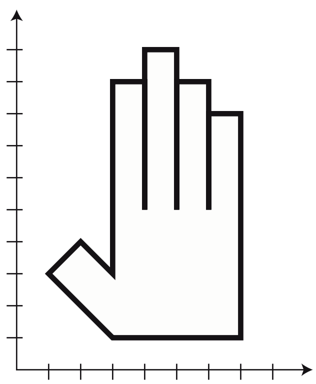


This hand consists of 13 line segments, (3,1)-(1,3); (1,3)-(2,4); (2,4)-(3,3); (3,3)-(3,9); (3,9)-(4,9); (4,5)-(4,10); (4,10)-(5,10); (5,5)-(5,10); (5,9)-(6,9); (6,5)-(6,9); (6,8)-(7,8); (7,8)-(7,1); (7,1)-(3,1)

1. Find a normal to each line segment
2. Choosing the line segments in any order, build a BSP tree for this hand. If you need to split any line segments and **are confident in your ability** to compute the intersection point, you may eyeball the intersections.
3. Traverse the BSP tree for a view from (0,0). Compute the vectors and dot products for each tree node to decide which branch to traverse first. Label the edges in the diagram below with their rendering order (1, 2, 3, …).
4. Traverse the BSP tree for a view from (10,10). Once again, explicitly compute the vectors and dot products for each tree node and label the edges with their rendered order.



Part 3



Part 4

CMSC 435/634 Solution Set 1

1. Find a normal to each line segment

Each edge is

e = v1 – v0 = (x1,y1) – (x0,y0) = (x1-x0, y1-y0) = (dx, dy)

For the normal to that edge, can either use 2D normal = (-dy, dx) or 3D cross product with (0,0,1):

(0, 0, 1) $\times$ (dx, dy, 0) = (0*0-1*dy, 1*dx-0*0, 0*dx-0*dy) = (-dy, dx, 0)

The negation of these are also OK: (dy, -dx) instead of (-dy, dx), as is any uniform scaling: (-k*dy, k*dx). Full list:

(3,1)-(1,3) $\Rightarrow$ (-2,-2)

(1,3)-(2,4) $\Rightarrow$ (-1,1)

(2,4)-(3,3) $\Rightarrow$ (1,1)

(3,3)-(3,9) $\Rightarrow$ (-6,0)

(3,9)-(4,9) $\Rightarrow$ (0,1)

(4,5)-(4,10) $\Rightarrow$ (-5,0)

(4,10)-(5,10) $\Rightarrow$ (0,1)

(5,5)-(5,10) $\Rightarrow$ (-5,0)

(5,9)-(6,9) $\Rightarrow$ (0,1)

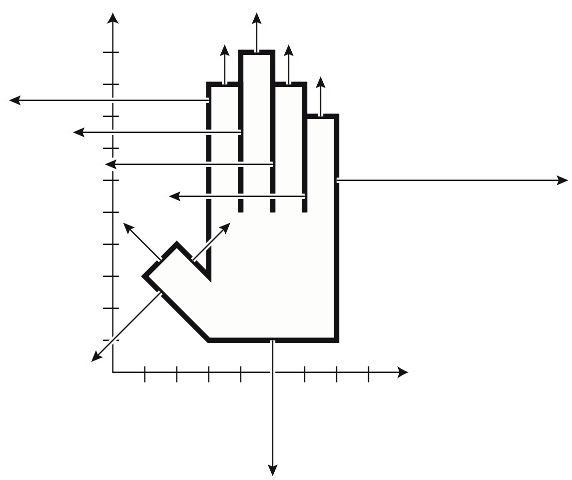
(6,5)-(6,9) $\Rightarrow$ (-4,0)

(6,8)-(7,8) $\Rightarrow$ (0,1)

(7,8)-(7,1) $\Rightarrow$ (7,0)

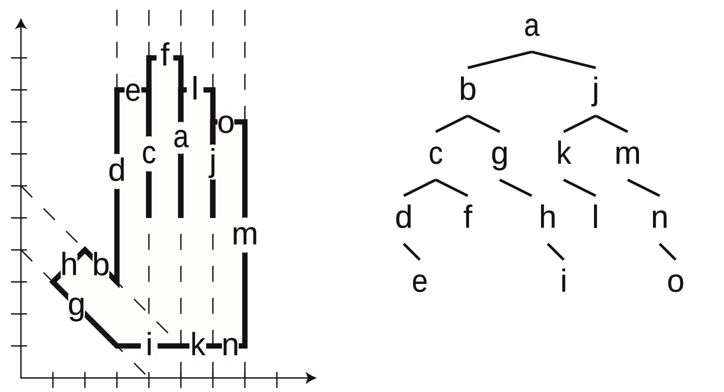
(7,1)-(3,1) $\Rightarrow$ (0,-4)

Shown on the hand, these normals are:



2. Choosing the line segments in any order, build a BSP tree for this hand. If you need to split any line segments and **are confident in your ability** to compute the intersection point, you may eyeball the intersections.

Pick an edge, e. Evaluate the end points of each other edge e_i against the plane equation for edge e. If both endpoints of e_i evaluate == 0, that edge e_i shares the BSP node with e. If both endpoints are ≥ 0, put e_i on the left branch of e. If both are ≤ 0, put e_i on the right side branch. If one is positive and one is negative, compute intersection (can use already computed plane equation values to do this) and put one of the new segments on each side of the tree.



Labels   Tree we ultimately construct

2.1 Start with a = (5,5)-(5,10) with normal (-5,0). Plane eq: n•p – n•v0 = -5*x + 5*5 = -5*x + 25

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(3,1)-(1,3)	-5*3+25 = 10	-5*1+25 = 20	Left
(1,3)-(2,4)	-5*1+25 = 20	-5*2+25 = 15	Left
(2,4)-(3,3)	-5*2+25 = 15	-5*3+25 = 10	Left
(3,3)-(3,9)	-5*3+25 = 15	-5*3+25 = 15	Left
(3,9)-(4,9)	-5*3+25 = 15	-5*4+25 = 5	Left
(4,5)-(4,10)	-5*4+25 = 5	-5*4+25 = 5	Left
(4,10)-(5,10)	-5*4+25 = 5	-5*5+25 = 0	Left
(5,9)-(6,9)	-5*5+25 = 0	-5*6+25 = -5	Right
(6,5)-(6,9)	-5*6+25 = -5	-5*6+25 = 0	Right
(6,8)-(7,8)	-5*6+25 = -5	-5*7+25 = -10	Right
(7,8)-(7,1)	-5*7+25 = -10	-5*7+25 = -10	Right
(7,1)-(3,1)	-5*7+25 = -10	-5*3+25 = 10	Split

The edge (7,1)-(3,1) spans edge a, so we need to compute an intersection point, this point is at t = plane_eq_at_v1 / (plane_eq_at_v1–plane_eq_at_v2) = -10/(-10 – 10) = 10/20 = .5

Intersection point is (7,1) + .5 ((3,1) – (7,1)) = (7,1) + (-2,0) = (5,1). For new segments:

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(7,1)-(5,1)	-5*7+25 = -10	-5*5+25 = 0	Right
(5,1)-(3,1)	-5*5+25 = 0	-5*3+25 = 10	Left

2.2 Repeat with b = (2,4)-(3,3) with normal (1,1) using only edges marked Left by a. Plane equation is x + y – (2+4) = x + y - 6

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(5,1)-(3,1)	5+1-6 = 0	3+1-6 = -2	Right
(3,1)-(1,3)	3+1-6 = -2	1+3-6 = -2	Right
(1,3)-(2,4)	1+3-6 = -2	2+4-6 = 0	Right
(3,3)-(3,9)	3+3-6 = 0	3+9-6 = 6	Left
(3,9)-(4,9)	3+9-6 = 6	4+9-6 = 7	Left

(4,5)-(4,10)	4+5-6 = 3	4+10-6 = 8	Left
(4,10)-(5,10)	4+10-6 = 8	5+10-6 = 9	Left

2.3 Repeat with c = (4,5)-(4,10) with normal (-5,0) using only edges marked Left by b. Plane equation is -5 x – (-5*4) = -5 x + 20

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(3,3)-(3,9)	-5*3+20 = 5	-5*3+20 = 5	Left
(3,9)-(4,9)	-5*3+20 = 5	-5*4+20 = 0	Left
(4,10)-(5,10)	-5*4+20 = 0	-5*5+20 = -5	Right

2.4 Repeat with d = (3,3)-(3,9) with normal (-6,0) using only edges marked Left by c. Plane equation is -6 x – (-6*3) = -6 x + 18

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(3,9)-(4,9)	-6*3+18 = 0	-6*4+18 = -6	Right

2.5 e = (3,9)-(4,9) is the only one left in this branch, so it is a BSP leaf node

2.6 Moving back up, f = (4,10)-(5,10) is the only one left in c's right branch, so it is a BSP leaf node.

2.7 g, h and i are all on b's right side. Start with g = (3,1)-(1,3) with normal (-2,-2). Plane equation is –2 x –2 y – (-2*3 –2*1) = -2 x –2 y + 8

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(5,1)-(3,1)	-2*5-2*1+8 = -4	-2*3-2*1+8 = 0	Right
(1,3)-(2,4)	-2*1-2*3+8 = 0	-2*2-2*4+8 = -4	Right

2.8 Repeat with h = (1,3)-(2,4) with normal (-1,1) using only edges marked right by g. Plane equation is –x + y – (-1+3) = -x + y - 2

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(5,1)-(3,1)	-5+1-2 = -6	-3+1-2 = -4	Right

2.9 i is the only one left in this branch, so it is a BSP leaf node.

2.10 Now we move all the way back up to the right children of a: j, k, l, m, n and o. Start with j = (6,5)-(6,9) with normal (-4,0). The plane equation is –4 x – (-4*6) = -4 x + 24

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(5,9)-(6,9)	-4*5+24 = 4	-4*6+24 = 0	Left
(6,8)-(7,8)	-4*6+24 = 0	-4*7+24 = -4	Right
(7,8)-(7,1)	-4*7+24 = -4	-4*7+24 = -4	Right
(7,1)-(5,1)	-4*7+24 = -4	-4*5+24 = 4	Split

(7,1)-(5,1) is split by edge j. It is split at t = -4 / (-4-4) = .5. That corresponds to a point of (7,1) + .5 ((5,1)-(7,1)) = (7,1) + .5 (-2,0) = (7,1) + (-1,0) = (6,1). For the new segments:

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(7,1)-(6,1)	-4*7+24 = -4	-4*6+24 = 0	Right
(6,1)-(5,1)	-4*6+24 = 0	-4*5+24 = 4	Left

2.11 Of the edges left of j, we pick k = (6,1)-(5,1) with normal (0,-4) if we reuse the previously computed normal or (0,-1) if we compute a new one for this segment. The plane equation is –4 y – (-4*1) = -4 y + 4

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(5,9)-(6,9)	-4*9+4 = -32	-4*9+4 = -32	Right

2.12 l is the only edge left in this branch, so it is a BSP leaf node.

2.13 Moving back up to the right-hand children of j: m, n and o. We pick m = (7,8)-(7,1) with normal (7,0). The plane equation is 7 x – (7*7) = 7 x – 49.

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(7,1)-(6,1)	7*7-49 = 0	7*6-49 = -7	Right
(6,8)-(7,8)	7*6-49 = -7	7*7-49 = 0	Right

2.14 Repeat with n = (7,1)-(6,1) with normal (0,-4) if we reuse the original segment's normal or (0,-1) if we recomputed for this specific segment. Using (0,-4) the plane equation is –4 y – (-4*1) = -4 y + 4

Line segment	**Endpoint 1**	**Endpoint 2**	**Result**
(6,8)-(7,8)	-4*6+4 = -20	-4*7+4 = -24	Right

2.15 o is the only remaining edge in this branch, and the only remaining edge overall, so it is a BSP leaf node and the tree is complete.

3. Traverse the BSP tree for a view from (0,0). Compute the vectors and dot products for each tree node to decide which branch to traverse first. Label the edges in the diagram below with their rendering order (1, 2, 3, …).

At each node, evaluate n•v for p-v0 for p = (0,0) and v0 = one endpoint of the edge. This is exactly the same as evaluating the edge plane equation for p:

n•v = n•(p-v0) = n•p – n•v0 = plane equation

If this is negative, all of the left branch will be further from us than the edge e, and all of the right branch will be closer. Traverse the left branch first, then render the node, then traverse the right branch. If it is positive the reverse is true: traverse the right branch, then render the node, then traverse the left branch. If it is 0, we are viewing edge e edge-on, so the left and right branches are separated on the screen by e and can be rendered in either order.

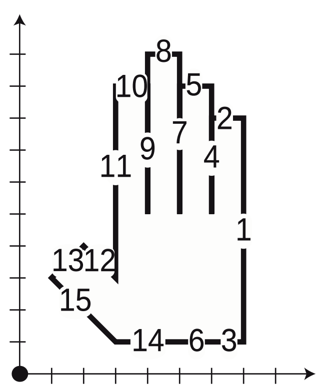
Segment	**Plane Equation**	**Evaluated at (0,0)**	**Traversal order**
a	-5 x + 25	25	Right, a, Left
j	-4 x + 24	24	Right, j, Left
m	7 x – 49	-49	Left, m, Right
n	-4 y + 4	4	Right, n, Left
o	y – 8	-8	Left, o, Right
k	-4 y + 4	4	Right, k, Left
l	y – 9	-9	Left, l, Right
b	x + y – 6	-6	Left, b, Right
c	-5 x + 20	20	Right, c, Left
f	x – 10	-10	Left, f, Right
d	-6 x + 18	18	Right, d, Left

Segment	Plane Equation	Evaluated at (0,0)	Traversal order
e	x – 9	-9	Left, e, Right
g	-2 x – 2 y + 8	8	Right, g, Left
h	-x + y – 2	-2	Left, h, Right
i	-4 y + 4	4	Right, i, Left

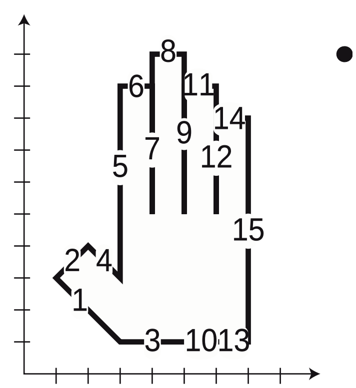
4. Traverse the BSP tree for a view from (10,10). Once again, explicitly compute the vectors and dot products for each tree node and label the edges with their rendered order.

Now repeat the same equations using p = (10,10)

Segment	Plane Equation	Evaluated at (0,0)	Traversal order
a	-5 x + 25	-25	Left, a, Right
b	x + y – 6	14	Right, b, Left
g	-2 x – 2 y + 8	-32	Left, g, Right
h	-x + y – 2	-2	Left, h, Right
i	-4 y + 4	-36	Left, i, Right
c	-5 x + 20	-30	Left, c, Right
d	-6 x + 18	-42	Left, d, Right
e	x – 9	1	Right, e, Left
f	x – 10	0	Either, f, Either
j	-4 x + 24	-16	Left, j, Right
k	-4 y + 4	-36	Left, k, Right
l	y – 9	1	Right, l, Left
m	7 x – 49	21	Right, m, Left
n	-4 y + 4	-36	Left, n, Right
o	y – 8	2	Right, o, Left



Part 3



Part 4