OBJECT SPACE METHODS

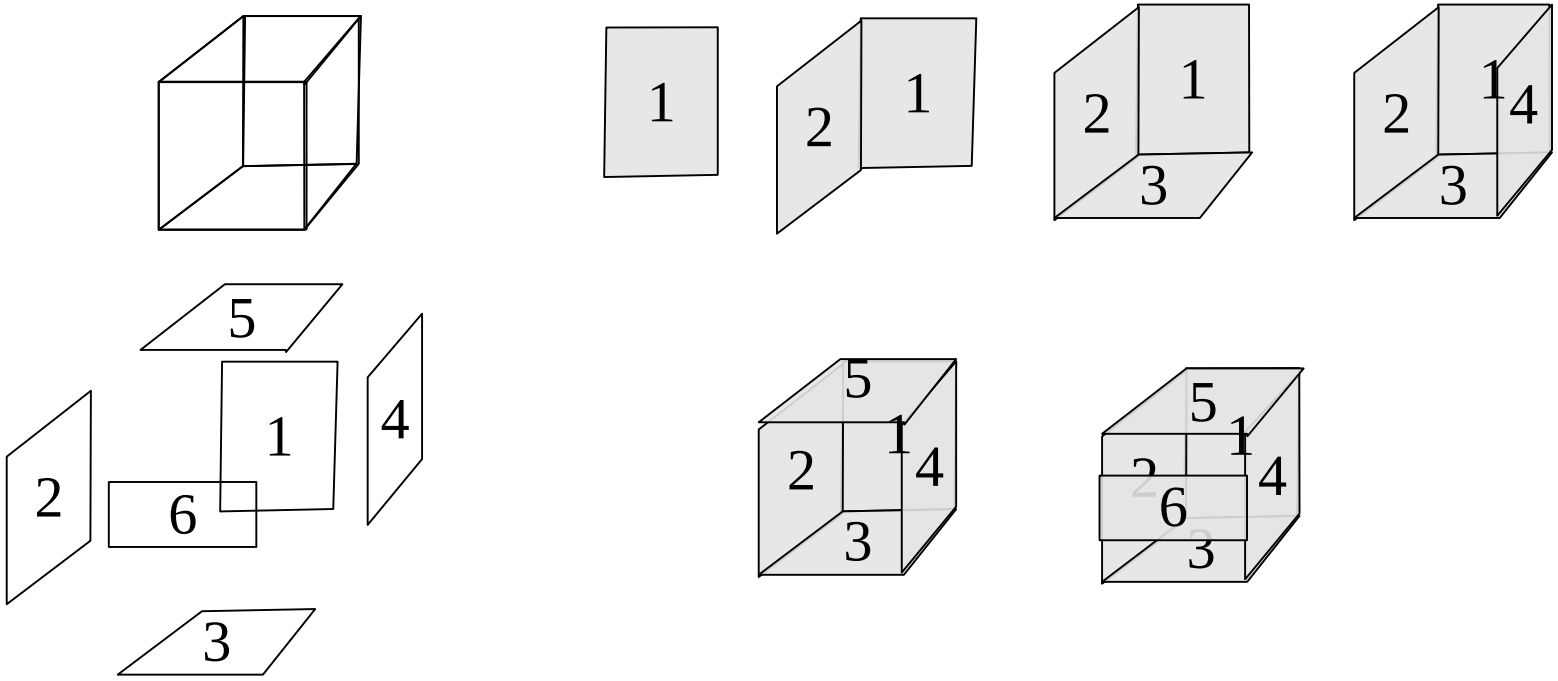
Amrita Kaur, Assistant Professor
Sushma Jain, Associate Professor
Anupam Garg, Assistant professor
Santarpal Singh, Assistant Professor

Outline

- Concept of Depth sorting
- Depth sorting algorithm

Depth Sorting

Also known as painters algorithm. First draw the distant objects than the closer objects. Pixels of each object overwrites the previous objects.



Depth-sort algorithm

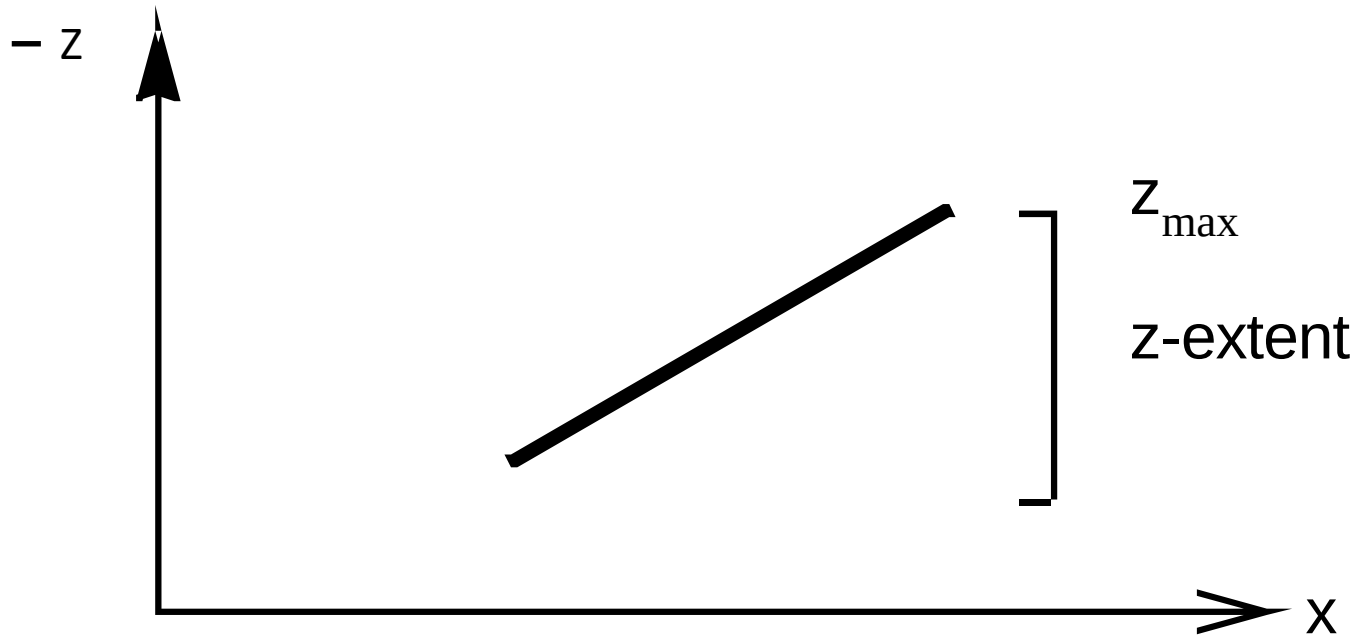
- The idea here is to go back to the front drawing all the objects into the frame buffer with nearer objects being drawn over top of objects that are further away
- Simple algorithm:

- Sort all the polygons based on their farthest z-coordinate
- Resolve ambiguities
- Draw the polygons in order from back to front

- This algorithm would be very simple if the z coordinates of the polygons were guaranteed never to overlap. Unfortunately that is usually not the case, which means that step 2 can be somewhat complex

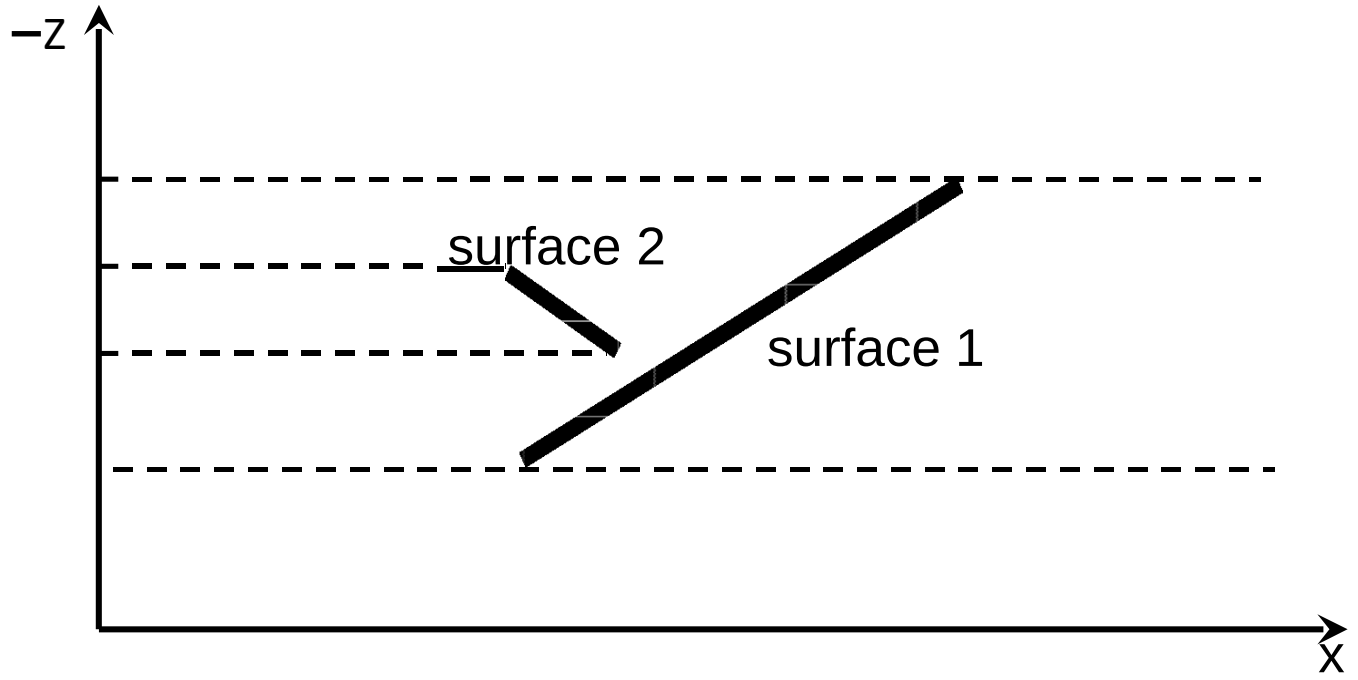
Depth-sort algorithm

- First must determine z-extent for each polygon

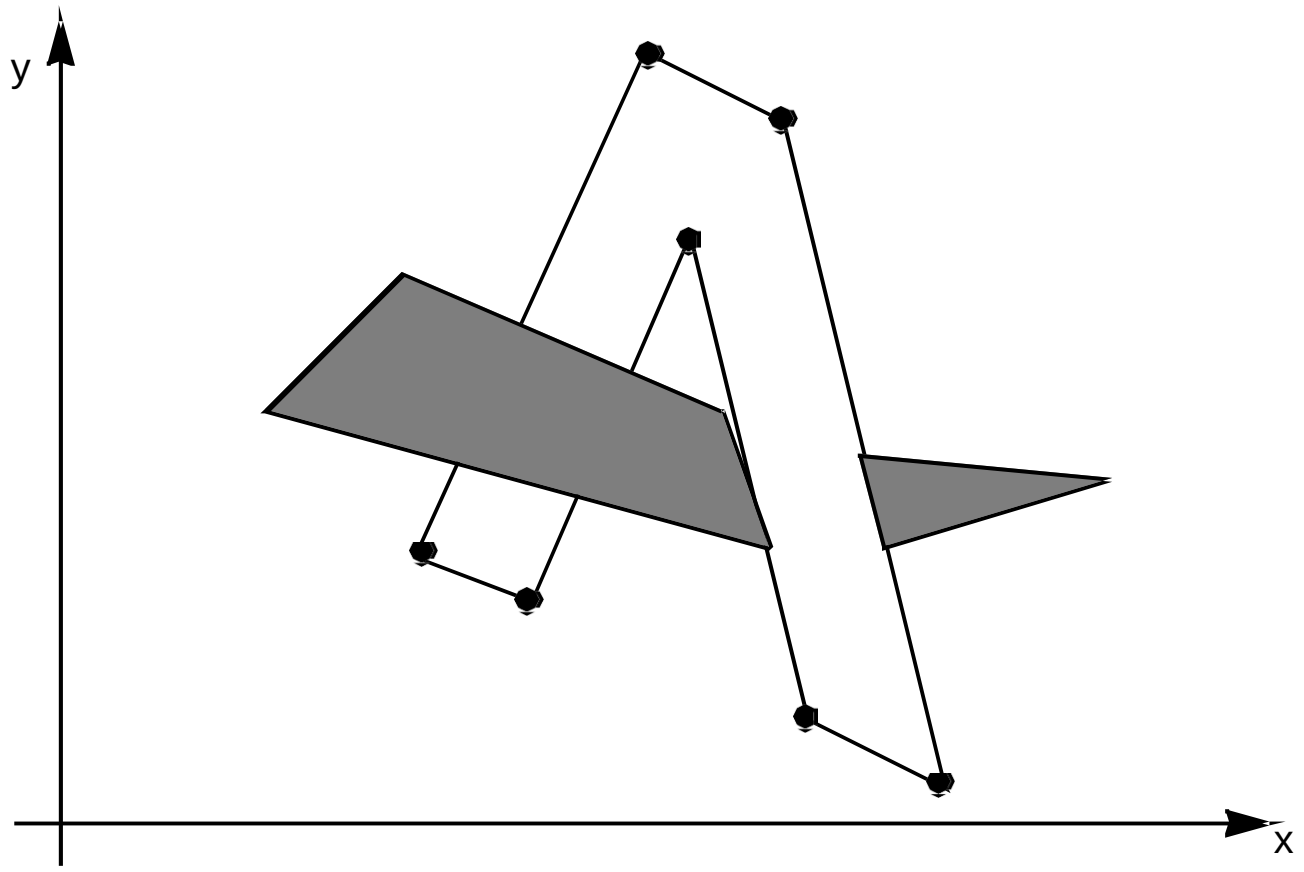


Depth-sort algorithm

- Ambiguities arise when the z-extents of two surfaces overlap.



Depth-sort algorithm



Depth-sort algorithm

- All polygons whose z extents overlap must be tested against each other.

- We start with the furthest polygon and call it P. Polygon P must be compared with every polygon Q whose z extent overlaps P's z extent. 5 comparisons are made. If any comparison is true then P can be written before Q. If at least one comparison is true for each of the Qs then P is drawn and the next polygon from the back is chosen as the new P.

Depth-sort algorithm

1. Do P and Q's x-extents not overlap?
2. Do P and Q's y-extents not overlap?
3. Is P entirely on the opposite side of Q's plane from the viewport?
4. Is Q entirely on the same side of P's plane as the viewport?
5. Do the projections of P and Q onto the (x,y) plane not overlap?

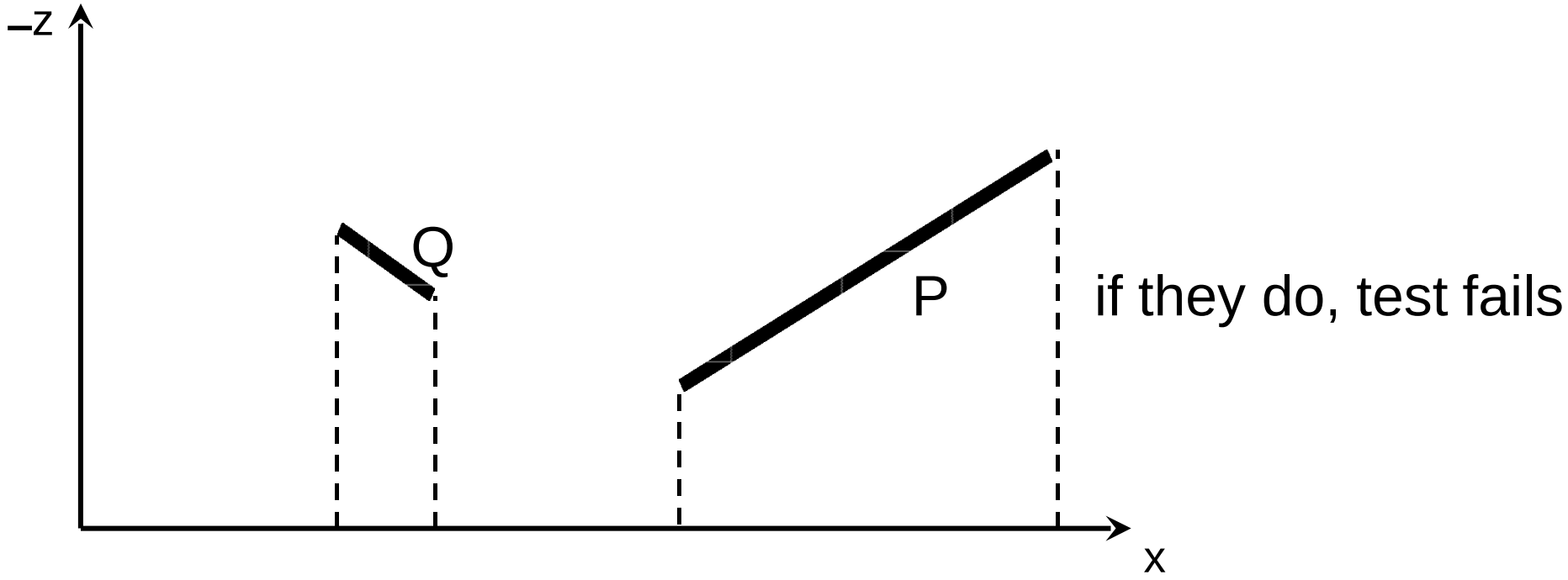
- If all 5 tests fail we quickly check to see if switching P and Q will work. Tests 1, 2, and 5 do not differentiate between P and Q but 3 and 4 do. So we rewrite 3 and 4 as:

  3’. Is Q entirely on the opposite side of P's plane from the viewport?

  4’. Is P entirely on the same side of Q's plane as the viewport?

Depth-sort algorithm

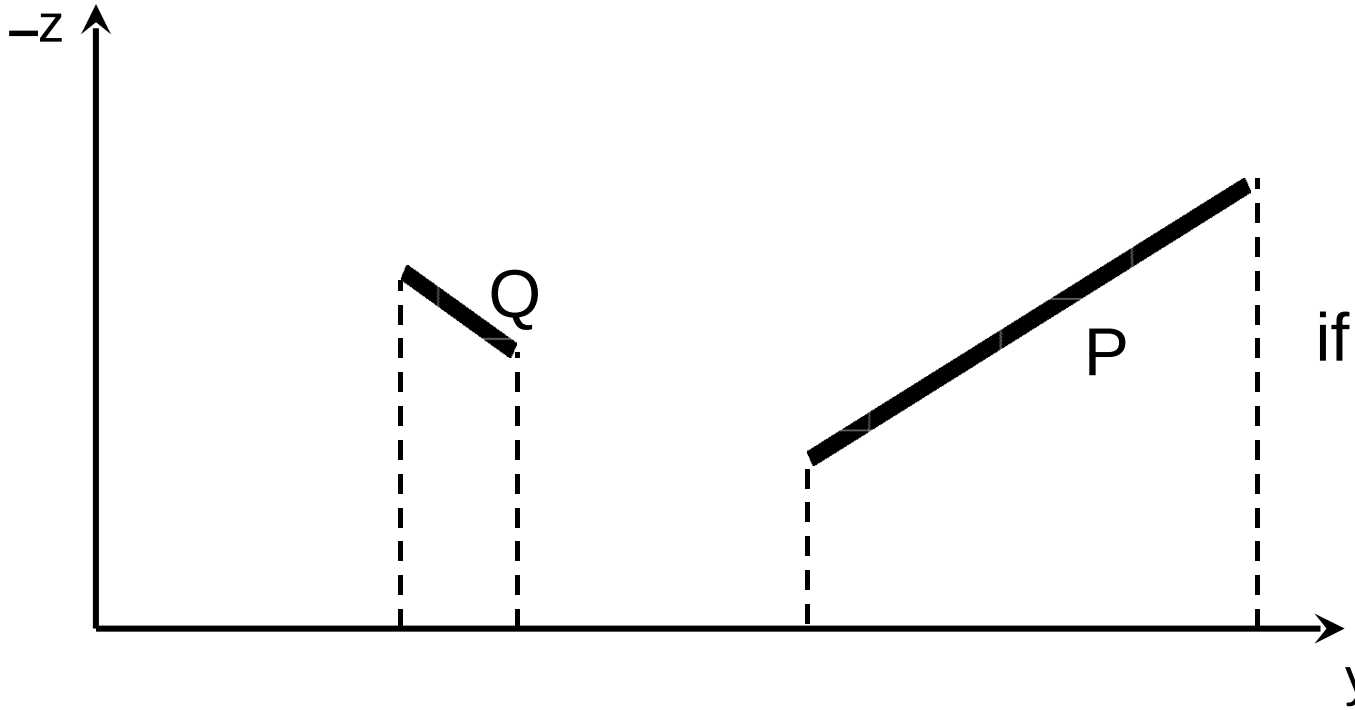
x - extents not overlap?



if they do, test fails

Depth-sort algorithm

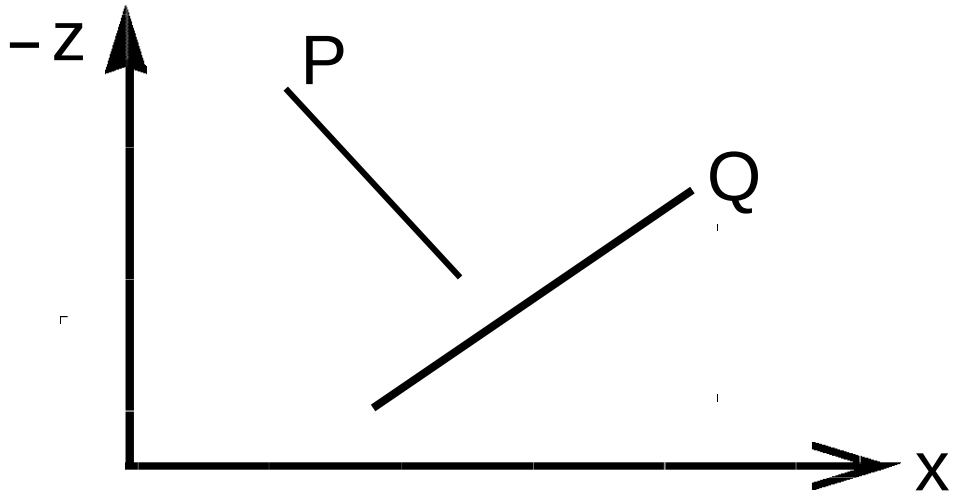
y - extents not overlap?



if they do, test fails

Depth-sort algorithm

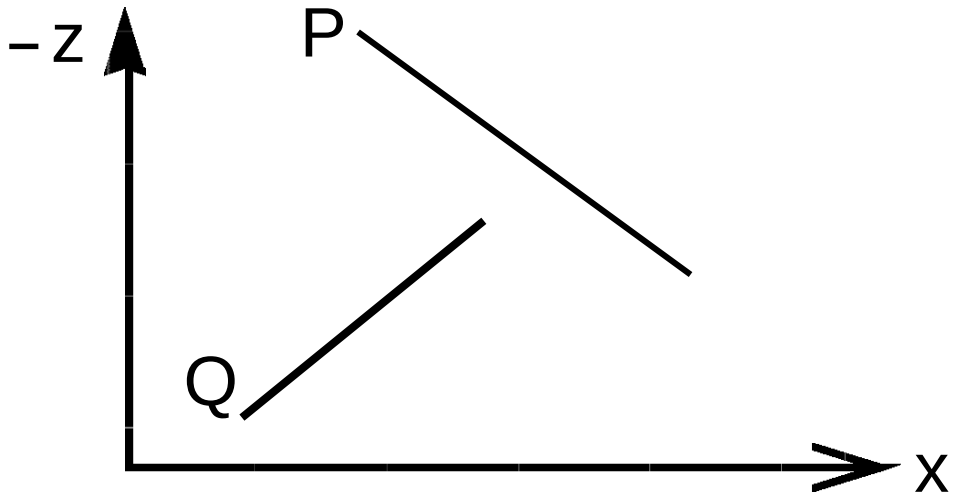
Is P entirely behind the surface Q relative to the viewing position (i.e., behind Q's plane with respect to the viewport)?



Test is true…

Depth-sort algorithm

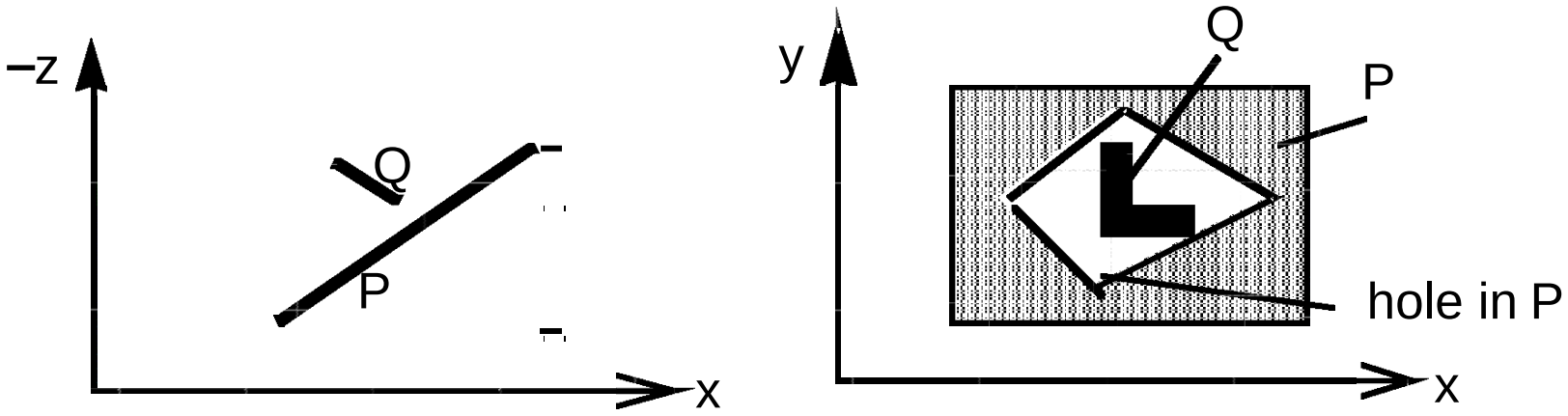
Is Q entirely in front of P's plane relative to the viewing position (i.e., the viewport)?



Test is true…

Depth-sort algorithm

Do the projections of P and Q onto the (x,y) plane not overlap?



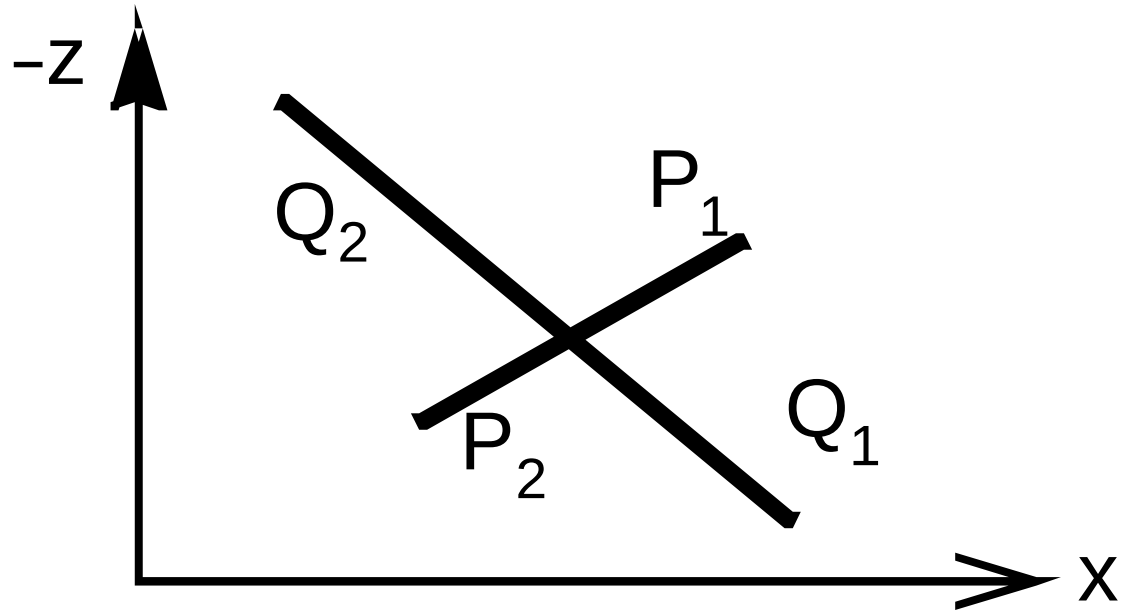
Test is true…

Depth-sort algorithm

- If all tests fail…
  - … then reverse P and Q in the list of surfaces sorted by maximum depth
  - set a flag to say that the test has been performed once.
  - If the tests fail a second time, then it is necessary to split the surfaces and repeat the algorithm on the 4 new split surfaces

Depth-sort algorithm

- Example:
  - We end up processing with order Q2,P1,P2,Q1



THANKING YOU