Clipping polygons — the Sutherland-Hodgman algorithm
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex.
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list
**Clipping polygons — the Sutherland-Hodgman algorithm**

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

**Case 1** — vertex B is added to the output list

**Case 2** — vertex B’ is added to the output
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list
Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)

```
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list

Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)

Case 3 — no vertex added
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list
Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)
Case 3 — no vertex added (segment AB clipped out)
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list

Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)

Case 3 — no vertex added (segment AB clipped out)

Case 4 — vertices A’ and B are added to the output
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,

Case 1 — vertex B is added to the output list

Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)

Case 3 — no vertex added (segment AB clipped out)

Case 4 — vertices A’ and B are added to the output (edge AB is clipped to A’B)
Clipping polygons — the Sutherland-Hodgman algorithm

Clipping polygons would seem to be quite complex. A single polygon can actually be split into multiple polygons (can you draw an example?).

The Sutherland-Hodgman algorithm clips a polygon against all edges of the clipping region in turn.

The algorithm steps from vertex to vertex, adding 0, 1, or 2 vertices to the output list at each step.

Assuming vertex A has already been processed,
Case 1 — vertex B is added to the output list
Case 2 — vertex B’ is added to the output (edge AB is clipped to AB’)
Case 3 — no vertex added (segment AB clipped out)
Case 4 — vertices A’ and B are added to the output (edge AB is clipped to A’B)
Typically, this is coded recursively — the recursive implementation lends itself to efficient hardware implementation.
Typically, this is coded recursively — the recursive implementation lends itself to efficient hardware implementation.

This algorithm is actually quite general — the clip region can be any convex polygon in 2D, or any convex polyhedron in 3D.
Typically, this is coded recursively — the recursive implementation lends itself to efficient hardware implementation.

This algorithm is actually quite general — the clip region can be any convex polygon in 2D, or any convex polyhedron in 3D.