Question Midterm–1: (Solution, p 2) Explain a reason why the painter's algorithm for handling occlusion, in which objects are drawn starting from the farthest away, does not work well for general 3D rendering.

Question Midterm–2: (Solution, p 2) What makes mipmaps useful for efficiently computing texture mappings?

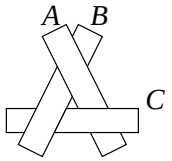
Question Midterm–3: (Solution, p 2) Define the term *anisotropic filtering*.

Question Midterm–4: (Solution, p 2) How is OpenGL's stenciling feature useful for computing reflections?

Question Midterm–5: (Solution, p 2) Describe how Phong's model would compute specular reflection for a point where the light source direction is $\mathbf{s}$, the surface normal vector is $\mathbf{m}$, and the direction to the user is $\mathbf{v}$. You can assume that all of these vectors are unit vectors.

Solution Midterm–1: (Question, p 1) There are two major shortcomings.

- Objects cannot always be sorted in a strict ordering. For example, we might have three logs in an arrangement where A overlaps with B, B with C, and C with A.



- Determining whether one object is in front of another is is complex and inefficient. Moreover, sorting the objects takes more than $O(n)$ time (even if it is only $O(n \log n)$ time).

[This doesn't mean that the painter's algorithm is irrelevant. It can make sense in particular applications, like in drawing a ball-and-stick model of a molecule, where the model has special properties that make sorting easy.]

Solution Midterm–2: (Question, p 1) When a textured surface is far away, many texels can end up mapping to the same pixel. Averaging over all the texels a pixel overlaps is inefficient. With mipmaps being reduced versions of the original texture, the texels are effectively pre-averaged, allowing for efficiently estimating the proper color for a pixel.

Solution Midterm–3: (Question, p 1) Anisotropic filtering involves pre-computing reduced versions of a texture at different aspect ratios. When the user looks at a textured surface at an oblique angle, then, the appropriately reduced texture will already be available.

Solution Midterm–4: (Question, p 1) For each mirrored surface in a scene, a renderer can compute a stencil representing the pixels to which the mirrored surface corresponds. Using this stencil, it can move the viewpoint to the camera's location after being reflected through the mirror, and re-render the scene as it would be seen from the mirror's other side. The stencil will prevent this second rendering (and following renderings, if the scene has multiple mirrors) from overwriting values computed during the first rendering.

Solution Midterm–5: (Question, p 1) According to Phong's model, the specular reflection would be $(\mathbf{r} \cdot \mathbf{v})^f$, for some exponent f that varies according to the material. This uses the vector $\mathbf{r}$, representing the reflection of $\mathbf{s}$ through $\mathbf{m}$, which can be computed as $2(\mathbf{m} \cdot \mathbf{s})\mathbf{m} - \mathbf{s}$.