D1 Computer Graphics (25 points)

(a) (10 points) Compare and contrast Bresenham's (aka "midpoint") 2-D line drawing algorithm with the DDA algorithm in terms of efficiency. Use as much algorithmic detail as possible.

(b) (10 points) Outline an efficient scan-line approach to filling simple n-sided 2-D polygons. Explain data structures used and give relevant functions in pseudocode.

(c) (5 points) How would your algorithm from **(b)** be affected if you knew that all of the polygons were *convex*? If you knew that they were *triangles*?

D2 Computer Graphics (25 points)

(a) (10 points) Consider a scene which consists of a light source L, the camera or eye E, a number of purely specularly reflective or refractive objects each represented by S, and a number of purely diffuse objects each represented by D. Light may travel from L to E along any number of paths, but basic ray tracing can only simulate a subset of them.

Using regular expression notation (such as x^+ means "one or more of x" and x^* means "zero or more of x") on the variables L, E, S, and D, show which light paths basic ray tracing can handle and explain why.

Briefly discuss an extension to ray tracing to extend the set of paths which can be simulated.

(b) (7 points) In ray tracing, once we have determined where a ray strikes an object, the illumination at the intersection point can be approximated with the Phong approach using the formula:

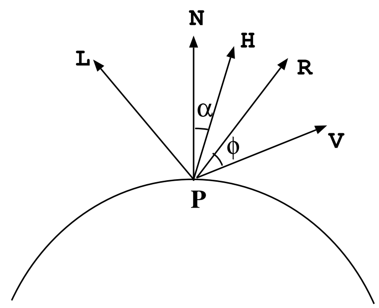
$$I = I_a k_a + \sum_i I_i k_d (\mathbf{l}_i \cdot \mathbf{n}) + \sum_i I_i k_s (\mathbf{r}_i \cdot \mathbf{v})^n \tag{1}$$

Explain what real visual effect each of the three terms is trying to model and what each of the following symbols means within the context of this formula: $I, I_a, i, I_i, k_a, k_d, k_s, \mathbf{l}_i, \mathbf{n}, \mathbf{r}_i, \mathbf{v}$, and n.

(c) (8 points) A disc is a finite, planar, circular object. Describe an algorithm to find the point of intersection of an arbitrary ray with an arbitrary disc in three dimensions. Make sure that you describe the parameters used to define both the ray and the disc.

D3 Computer Graphics (25 points)

(a) In the below figure, **N** is unit normal vector, **L** is unit direction vector to a point source, **V** is the unit vector pointing to the viewer, **R** is specular reflection direction, and **H** is the halfway vector. A somewhat simplified Phong specular-reflection model is obtained by replacing the dot product between view vector (**V**) and reflection vector (**R**) with the dot product between halfway vector (**H**) and unit normal (**N**) of a given point. **H** is the half way vector between **L** and **V**.



(i) (5 points) Show how the unit halfway vector can be calculated? Why is the above approximation done?

(ii) (4 points) Discuss the differences you might expect to see in the appearance of specular reflections modeled with $(\mathbf{N}\cdot\mathbf{H})^{n_s}$ compared to specular reflections modeled with $(\mathbf{V}\cdot\mathbf{R})^{n_s}$.

(b) (8 points) Say you have just installed a rendering program for shading poygons. Assuming that you have no way of finding whether the program uses flat, Gouraud, or Phong shading, devise a series of tests that would determine which shading method is used by the software. You may define a particular scene, lighting, viewing position and direction, etc.

(c) (8 points) Briefly describe the aspect of light modeled by radiosity techniques. Although computing a radiosity solution is as slow (or slower) than ray tracing a single image, what advantage does radiosity have over raytracing?

D4 Computer Graphics (25 points)

Hidden Surface Removal

(c) (9 points) (i) The Painter's algorithm for hidden surface removal is not an *on-line* algorithm, but z-buffer is. What do we mean by an *on-line* algorithm for hidden surface removal? (ii) Between Painter's and z-buffer, which algorithm has a better ability to incorporate transparency? (iii) What modification can be used to the other algorithm to incorporate transparency?

Transformations

(a) (8 points) (i) State which transformation is commutative: successive translations, successive rotations, successive scaling? (ii) Write properties of a rotation matrix.

(b) (8 points) Consider a 2D cartesian coordinate system. Given a point p, two perpendicular unit vectors $\vec{v}$ and $\vec{w}$, and two scale factors a and b, suppose we want to perform a non-uniform scale about p by a in direction $\vec{v}$ and b in direction $\vec{w}$. Give the appropriate matrix product to achieve this transformation.