

For this homework, please read Handout 2.

1. Consider the *winged-edge* representation of meshes described in Handout 2. Let's assume that the polygons in the mesh are all triangles (*i.e.*, that it is a trimesh).
 - (a) Write a function (in pseudocode) that returns the unit normal of a face.
 - (b) Write pseudocode that when given a direction vector (*i.e.* a vector pointing toward a viewer or a light) and a silhouette edge (w.r.t. the direction vector), walks the loop of edges that defines the mesh's silhouette.
2. Consider a sphere tree with the following representation:

```
struct SphereTree {  
    vec3f      center;    // center of the sphere  
    float     radius;    // radius  
    float     radius2;   // radius*radius  
    SphereTree *child1;  // children, which are contained in this  
    SphereTree *child2;  // sphere (or nullptr if this is a leaf)  
};
```

- (a) Sketch a function that given two sphere trees, `st1` and `st2`, creates a new sphere tree node with minimum radius that has `st1` and `st2` as its children.
- (b) Sketch an **efficient** function that when given a ray and a sphere tree, returns true if a ray intersects any of the leaves of the tree. Note that you will first need to define a function that tests ray-sphere intersection.