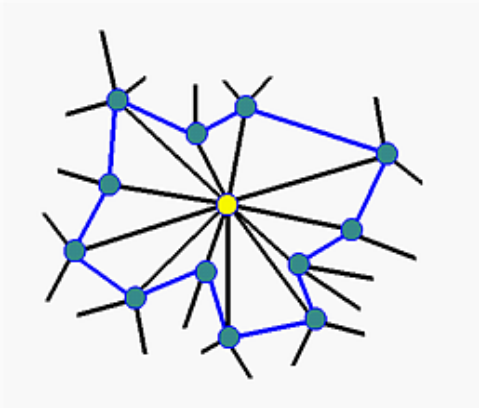
CS3621 Introduction to Computing with Geometry
Drill Exercises – The Winged-Edge Structure

Suppose the winged-edge data structure is used for representing a polyhedron of V vertices, E edges, and F facets. Answer the following questions:

1. What is the memory consumption? You can assume one word for each integer or pointer.
2. Given a vertex v, develop an algorithm that lists all adjacent edges of v in clockwise or counter-clockwise order. An edge is adjacent to v if that edge has v as one of its vertices. How many comparisons are required (in worst case)? Compare this against the data structure in the previous drill exercise.
3. Given a vertex v, develop an algorithm that lists all adjacent facets of v in clockwise or counter-clockwise order. A facet is adjacent to v if that facet has v as its vertex. How many comparisons are required (in worst case)? Compare this against the data structure in the previous drill exercise.
4. Given an edge e, develop an algorithm that lists all adjacent facets of e. A facet is adjacent to e if that facet has e as its edge. How many comparisons are required (in worst case)? Compare this against the data structure in the previous drill exercise.
5. Given a facet f, develop an algorithm that lists all adjacent facets of f in clockwise or counter-clockwise order. A facet is adjacent to f if it has a common edge or a common vertex with facet f. How many comparisons are required (in worst case)? Compare this against the data structure in the previous drill exercise.
6. Given a facet f, develop an algorithm that lists all of f's vertices (or edges) in clockwise or counter-clockwise order. How many comparisons are required (in worst case)? Compare this against the data structure in the previous drill exercise.
7. Given two vertices v_1 and v_2, not necessary adjacent, of a facet f. Develop an algorithm that lists all vertices between v_1 and v_2 in clockwise or counter-clockwise order. How many comparisons are required?
8. The concepts of *star* and *link* are very important in polyhedron modeling and editing. Suppose a polyhedron's facets are *all* triangles. Given a vertex, the yellow one in the figure below, the *star* of this vertex consists of *all* triangles that are adjacent to this vertex listed in clockwise or counter-clockwise order. The *link* of this vertex consists of all vertices, listed in either clockwise or counter-clockwise order, that are on edges adjacent to but different from the given vertex. Note that once we have a properly ordered vertices list, we can connect them together to form a *link* as shown in blue in the figure below.



Develop an algorithm that accepts a vertex and reports the link (in either vertex or edge form) of this vertex in clockwise or counter-clockwise order. From this link information, you can easily generate the star. How many comparisons are required? Compare this against the data structure in the previous drill exercise.

9. Continue with the previous exercise. What if the facets are not all triangles as shown below? How many comparisons are required? Compare this against the data structure in the previous drill exercise.