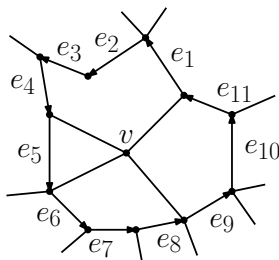


In-Class Exercise: Extra Credit

Given vertex v in a cell complex of a 2-manifold, the *link* of v is defined to be the edges that bound the faces that are incident to v , excluding the edges that are incident to v itself. Present a procedure (in pseudocode) that, given a vertex v of a DCEL, returns a list L consisting of the half edges of v 's link ordered counterclockwise about v . For example, in the figure below, a possible output would be $\langle e_1, \dots, e_{11} \rangle$. (Any cyclic permutation would be correct.)

**Solution:**

The solution provided below is very short, but a bit tricky. We start with any edge e that is directed out of v . We start following edges around the face lying to e 's left side, adding each to the link. (In the above figure, this will add e_1 through e_4 to the list, and the next edge visited will be directed into v .) When we return to v (that is, when the destination of the edge is v) we make a U-turn by setting e to its twin, and resume from there. (In the figure above, the next edge to be visited will be e_5 .)

```
link(Vertex v) {
    L = new empty-list
    e = e0 = v.incident; // any edge coming out of v
    do {
        e = e.next; // next edge about e's left face
        if (e.dest == v) // returning to v?
            e = e.twin;
        else
            add e to L; // e is an edge of the link
    } while (e != e0);
}
```

