HW2 – 327156998 - 204687339

1.

The exist(e) function checks whether the point e is already in the convex hull H. This is done by traversing the half-edge data structure and checking whether e is one of the endpoints of any of the edges. If e is found, the function returns true; otherwise, it returns false.

The processed(e) function checks whether the point e has already been processed by the algorithm. This is done by checking whether e is in the stack Q. If e is found in the stack, the function returns true; otherwise, it returns false.

The mark_processed(e) function marks the point e as having been processed by the algorithm. This is done by adding e to the stack Q.

During the course of the algorithm, the functions exist(e), processed(e), and mark_processed(e) are used to ensure that the algorithm only processes each point once, thereby ensuring a worst-case time complexity of Θ(n^2).

To perform the gift wrapping algorithm, the following steps are taken:

1. Choose an anchor point and add it to the convex hull H and the stack Q.
2. Find the point p that is farthest from the anchor point, and add it to the convex hull H and the stack Q.
3. While there are unprocessed points:

    a. Find the point q that is farthest from the line formed by the last two points added to the convex hull.

    b. If q is already in the convex hull (i.e., exist(q) returns true), stop the algorithm.

    c. If q has already been processed (i.e., processed(q) returns true), remove it from the stack Q and continue to the next iteration of the loop.

    d. Otherwise, add q to the convex hull H and the stack Q, and mark it as processed by calling mark_processed(q).

The convex hull H is now complete.

The gift wrapping algorithm has a worst-case time complexity of Θ(n^2) because each point is processed once, and the processing of each point involves finding the farthest point from a line, which takes Θ(n) time.

2.

a - A planar map is a graph that can be drawn on a plane without any of its edges crossing.

Every planar map can be represented as a collection of vertices, edges, and faces, where the faces are the regions bounded by the edges.

If a vertex has degree at least 4, then it is adjacent to at least 4 edges. These edges form a cycle around the vertex, and at least one of the edges in this cycle must be a boundary edge of the face containing the vertex. Thus, the face containing the vertex has degree at least 4.

On the other hand, if every vertex in the planar map has degree at most 3, then every face in the planar map has degree at most 3.

Therefore, a planar map must have either a vertex with degree at most 3 or a face with degree at most 3.

b - Every simple planar graph is a planar map, so it must have either a vertex with degree at most 3 or a face with degree at most 3.

If a simple planar graph has a face with degree at most 3, then it has a vertex with degree at most 3. This is because every face in a simple planar graph is bounded by at least 3 edges, and each of these edges is adjacent to at least one vertex. Therefore, the degree of a vertex in a simple planar graph is at least the degree of the face containing it.

If a simple planar graph has a vertex with degree at most 3, then it has a vertex with degree less than 6. This is because the degree of a vertex in a simple planar graph is equal to the number of edges incident to it, and the degree of a vertex cannot be negative.

Therefore, every simple planar graph has a vertex with degree less than 6.

4.

1. One construction for a simple polygon with 2k vertices that does not fully cover the polygon when guards are placed at every other vertex along the boundary is a regular k-gon with an additional line segment attached to one of its vertices. For example, a hexagon with guards placed at vertices 1, 3, and 5 does not fully cover the polygon because the line segment attached to vertex 2 is not visible to any of the guards. This construction can be generalized to any number of vertices by adding additional line segments to the regular k-gon.
2. One construction for a simple polygon with 3k vertices that does not fully cover the polygon when guards are placed at every third vertex along the boundary is a regular k-gon with k additional line segments attached to its vertices. For example, a nonagon with guards placed at vertices 1, 4, and 7 does not fully cover the polygon because the line segments attached to vertices 2 and 8 are not visible to any of the guards. This construction can be generalized to any number of vertices by adding additional line segments to the regular k-gon.
3. One construction for a simple polygon with n vertices that does not fully cover the polygon when guards are placed only at convex vertices is a regular (n-2)-gon with an additional line segment attached to one of its vertices and an additional point inside the polygon. For example, a hexagon with guards placed at its convex vertices (vertices 1 and 5) does not fully cover the polygon because the point inside the polygon is not visible to any of the guards. This construction can be generalized to any number of vertices by adding additional points inside the regular (n-2)-gon.