Amy Bui

1 Point Location: 2D

a. Translate the points of $S \cup \{q\}$ to the origin, where q is at the origin. For each point $p \in S$, draw vector $\overrightarrow{qp}$. For the set of vectors made about the origin at q, if 1) there exists two adjacent vectors that can make an obtuse angle with no other vector bisecting it, then q is outside $\mathcal{CH}(S)$; 2) if two adjacent vectors make 180^o angle that isn't bisect by another vector, then q is colinear to those two points and all three are on $\mathcal{CH}(S)$; and 3) all pairs of adjacent vectors make acute angles, then q is inside $\mathcal{CH}(S)$. This is done in linear time because we check adjacent vectors.

b. We know q is inside $\mathcal{CH}(S)$. Suppose we presorted S and know the points are ordered by x-coordinates, which itself takes $O(n \log n)$ time. Split the set of S across two half planes divided by the horizontal line going through q[1]. Take the left most and right most point in the upper plane and in the lower plane, and for each of those (at most) four points, make the vector from q to that extreme point. Pick three of those (at most) four points, and at least one of those triples will form a triangle Δxyz that contains q. Since we have at most $\binom{4}{3}$ triangles to check to check, this takes constant time to check this many point inclusions in a triangle. This works since q is already known to be within the convex hull; since we found (at most) 4 exteme points, these are in $\mathcal{CH}(S)$, and q will be contained in a face made by these vertices.

c. We know q is outside $\mathcal{CH}(S)$, so there exists some region "between" q and S that is empty and keeps them to opposite sides. We can find a support line of $\mathcal{CH}(S)$ that also keeps q to the opposite side of the line. Similar to part a, for each point $p \in S$, draw line segment $\overline{qp}$. We can find the shortest line segment in $O(n)$ time. Since this p is the "closest" to q out of all other points in S, itis some extreme point and so $p \in \mathcal{CH}(S)$. Draw the line perpendicular to this shortest $\overline{qp}$, and that is the support line that keeps S and q on opposite sides. No other points of S would be on the q-side of the line, because then such a point would closer to q than the closest point p, a contradition; additionally, since p is on the convex hull, this other closest point p' must also be on the convex hull; however, p' existance on the hull and on the q side of the line is a contradiction because it makes the polygon concave, and therefore not a convex hull.

d. Suppose we have calculated the $\mathcal{CH}(S)$, the three tasks can be done as:

part a It takes $O(\log n)$ time to do point inclusion with a convex polygon.

part b It takes $O(n)$ time to triangulate a convex hull. Supposing we store the triangulation in a DCEL, it can take $O(\log n)$ time to find the face in the DCEL that q belongs, and we report the three points of that triangle face, accessible in constant time.

part c If $\mathcal{CH}(S)$ is in a dynamic data structure, it takes $O(\log n)$ to find a bridge $\overline{qp}$ from q to $\mathcal{CH}(S)$, where $p \in \mathcal{CH}(S)$. If $\overline{qp} \Rightarrow y = mx + b$, the support line that separates S ad q is "tilted" $\overline{qp}$, i.e. $y = (m \pm \epsilon_1)x + (b \pm \epsilon_2)$, that includes point p.

[1] Diane had gone over this method in OH.

2 Point Location: 3D

a. For each $p \in S$ and q, translate the points about the origin, putting q at the origin. Using q, create three hyperplanes $H_{q,1}$, $H_{q,2}$, and $H_{q,3}$, that are orthogonal to the x_1-, x_2-, and x_3-axis, respectively. In $O(n)$ time, check if all points are to one side of a hyperplane. If at least one hyperplane keeps points of S to one side, then q is outside the $\mathcal{CH}(S)$; otherwise, q is inside the $\mathcal{CH}(S)$.

b. We know that q is inside the $\mathcal{CH}_{\text{3D}}(S)$. Sort the points of S in $O(n \log n)$ time, then find the at most 6 extreme points, leftmost and rightmost of each axis. Since these are extreme points, they are on $\mathcal{CH}_{\text{3D}}(S)$. We check at most the $\binom{6}{4}$ sets of four points that make a tetrahedral inside the hull to see if it contains q in $O(\log n)$ time. Report the set of four points that does contain q.

c. We know that q is outside the $\mathcal{CH}_{\text{3D}}(S)$. We take $O(n)$ time to find the 3 or 4 shortest $\overline{qp}$, for some $p \in S$. These points together form a face on the convex hull of S that is closest to q. A hyperplane made from these points form the support plane that separates q from S.

d. Suppose we have calculated the $\mathcal{CH}(S)$, the three tasks can be done:

part a Lay the 3D convex hull on the xy-plane, yz-plane, and xz-plane. The result is a triangulation in 2D plane of the 2D hull. We can do the three point inclusion for q in these planes in order to determine if q is in the hull in $O(\log n)$ time. If it is not in the hull in any of the 2D planes, it is not in the 3D hull.

part b Using linear programming we can find the tetrahedron containing q in linear time[2]. Laying the hull on a 2D plane gives the 2D hull and a triangulation. Select the three points of a triangle q is located in and the one point closets to q not on that face, and these four points are the four points of the tetrahedron that contain q in 3D.

part c Using linear programming we can find the hyperplane that separates q and S in linear time[2], by finding the facet of the convex hull of S that is closest to q. That facet turned into a plane is the support plane that separates q and S.

References

[1] Jake and Diane's office hours.

[2] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. 2008. Computational Geometry: Algorithms and Applications (3rd ed. ed.). Springer-Verlag TELOS, Santa Clara, CA, USA.

[2]Diane had gone over this in OH.