Dipen Kumar
Third-year undergraduate
Computer Science & Engineering
Indian Institute of Technology Delhi

Compute the convex hull for a set of n points in three-dimensional Euclidean space.
Implemented Gift wrapping algorithm
Complexity is O(nh) where n is the number of points in input set and h is the number of points in convex-hull

Sources -
http://wcipeg.com/wiki/Convex_hull
https://iq.opengenus.org/divide-and-conquer-convex-hull/
https://dccg.upc.edu/people/vera/wp-content/uploads/2014/11/GA2014-ConvexHulls3D-Roger-Hernando.pdf

Algorithm -

1) Project the 3D point in x-y plane
2) select an extreme point with the minimum x-coordinate and we know this will exist in convex-hull let's say this point1
3) We need another point connected to point1 in convex-hull in order to get an edge in the convex-hull.
   In order to get this another point I followed gift wrapping algorithm implemented in function findPoint2D in chull.py
   After we find point2 which together with point1 will form a line such that all other remaining points are only in one side among possible two sides in plane separated by the line joining point1 and point2.
4) Once we have a edge in convex-hull we find another point in given set of point to make a place such that all remaining points fall at one side in space separated by the plane.
5) When a new plane is created it also bring new edges to explore. Direction of edge determine the newly formed plane.
   Each edge is present in two planes with two different directions.
   Hence every edge is to be explored in both directions.
6) every time a new plane is created its edges are added in explored set and its edges in reversed form are added in pending to explore set which in this case is "edges"
   If its reversed form is already present in explored list then it is not added in pending list.
   Also those who are added in explored list are deleted from pending list.
7) continue finding new planes with an edge long as there exist any edge in the pending list.

Running Time Analysis -

1) Time to find new plane with given edge is "n" because it iterate over all remaining points and select the point with maximum possible angle.
2) new planes to find is of order O(h) because h is the number of points in convex-hull, and triangular faces in convex-hull i.e. new planes is proportional to number of points in convex-hull.
3) Addition deletion and searching in set is of order log(m) where m is the size of Set
4) Overall complexity is O(h(n+log(m))) where m is the size of set storing explored and pending edges which in worst case is of order n. Hence in worst case it becomes O(nh)

NOTE: My program runs correctly and find the set of the points of the vertexes of the minimum polyhedron or may provide points that are on the surface or edge of this convex-polyhedron with minimum volume. It run correctly for correct set of input.

How to run -

Run the chull.py with two command line argument where first argument is the path with filename of input file and second argument is the path with filename of output file

example steps -

1) change to directory which contains chull.py
2) python chull.py CONVEX.IN CONVEX.OUT

Python version used 3.7.6