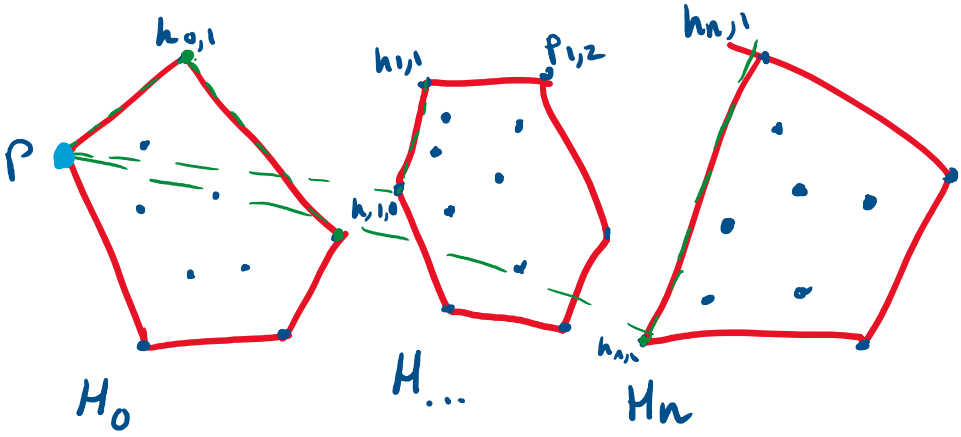
P4

Monday, February 2, 2026 6:38 PM

The h subsets have already been calculated, along with their convex hulls, stored in counterclockwise order, by definition of the convex hull problem.

We can use the leftmost point of all subset hulls to begin the search. This is O(n).

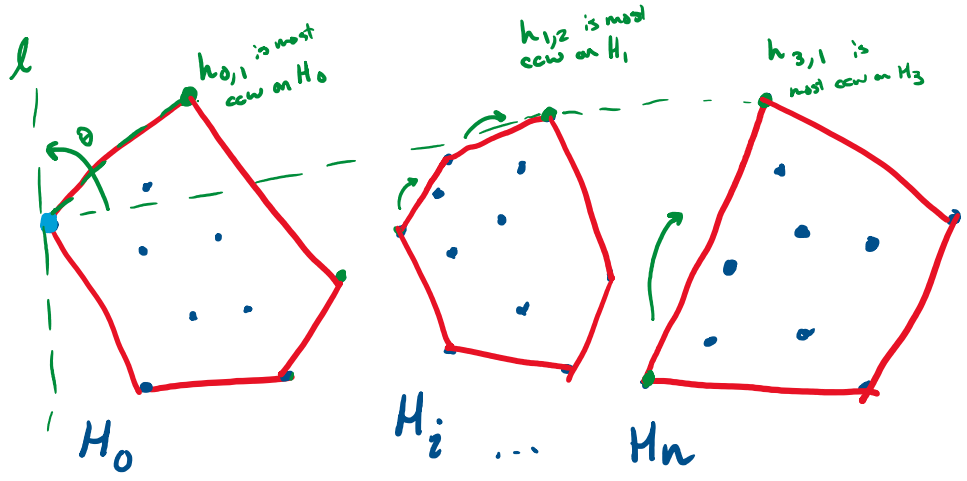
We will initialize pointers on the leftmost point of each hull. We will use the global leftmost point as our first point on the Convex Hull, called P.



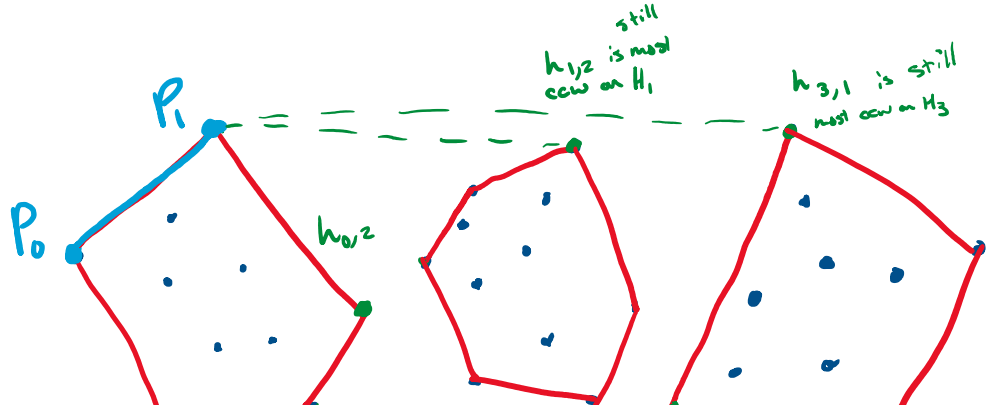
To determine the second point, we will use the vertical axis generated from P.

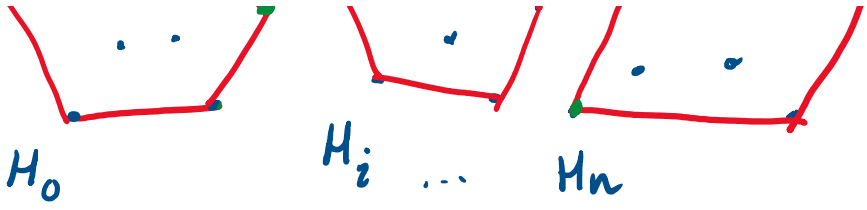
For each hull, H_i, we will step through the given ccw-sorted list given, starting at $h_{i,0}$, where $h_{i,0}$ represents the leftmost point on H_i. On H_0, we will start on $h_{0,1}$, since $P = h_{0,0}$ it is an edge case.

For each h_i pointer, we will step along h_i until the angle between $\overline{P h_{i,i}}$ and the vertical line from P is minimized.



Once all pointers cannot move, the best of all pointers is added to the convex hull.





We now check the pointers, this time evaluating $ccw(P_0, P_1, h_{i,i})$. We step along h_i until it cannot increase and choose the best from all pointers again. We repeat this until the convex hull is formed by closing itself.

Correctness:

We do not need to backtrack, unlike the monotone chain algorithm because if a better point at any given step could appear, the algorithm, would have stepped to it.

In other words, the next point is the correct global convex hull is always found in each stage because it checks all available hull points that could possibly [illegible] given step.

available hull points that could possibly be more ccw of the given step. This means there is no need to backtrack.

Time Complexity:

Because there is no need to backtrack, the algorithm walks along all points of all convex hulls in the worst case scenario only once. There can be at most n points on the subset convex hulls, therefore n points are walked once, so

$O(n)$.