How about maximal points in three dimensions?

Brute force in d dimensions will work correctly using \( O(d \cdot n^2) \).

Sweeping by a plane \( \Pi \) to \( Y-Z \) in decreasing \( x \) coordinates.

Obs.: Point having maximal \( x \)-coord is maximal, say \( \phi \).

When we visit the next point, say \( q \), we know \( x(q) < x(\phi) \).

\( y(q) : y(\phi) \quad z(q) : z(\phi) \)
In the generic step, the latest point visited, say \( r \), will not be maximal if and only if one of the points visited previously has a higher \( y \) coordinate and a higher \( z \) coordinate.

We want to test if \( r \) is inside or outside the staircase formed by the previously visited points.
1. Can we do it quickly even if staircase is large?
2. Can we update the staircase quickly?

Problem: Design an efficient data structure for the staircase so that 1 and 2 can be supported.
Given a balanced BST, can we support insertions, search and "arbitrary # of deletions" in $O(\log n)$ time?

In general, the following data structure operation on sorted sets are considered very useful:

1. Splitting out an interval of points
2. Concatenating two sorted intervals into a single interval

Concatenatable queues
In the staircase, we will delete the points one by one paying $O(\log n)$ cost per deletion.

One single iteration may be expensive (let a point be deleted) but overall each point can be inserted or deleted at most once.

$$\Rightarrow O(n \log n)$$ overall for 3D max.

Amortized Analysis