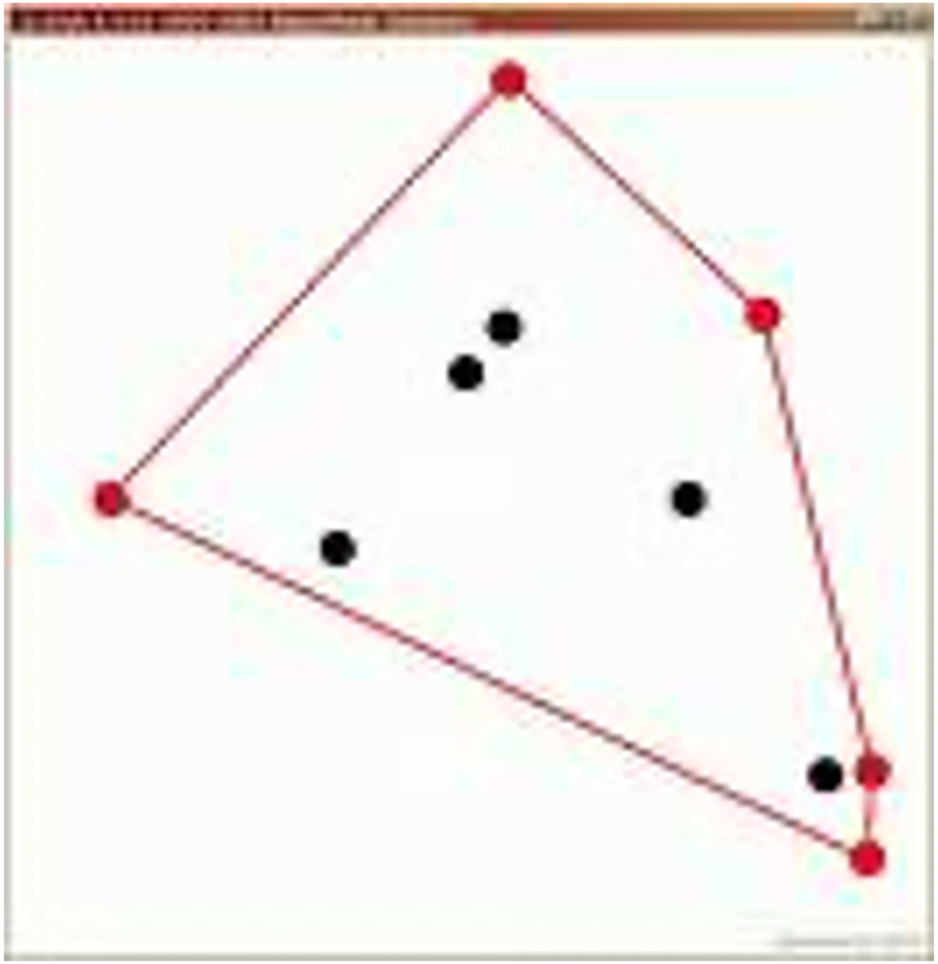
Convex Hulls

- Given: A set *Z* of *n* points in the plane.
- Find: Smallest convex set containing *Z*.



Some Definitions

- A set S in the plane is convex iff for every pair of points p_1 and p_2 in S, the line-segment p_1p_2 is in S.
- A point p in a convex set S is said to be extreme (or a corner) iff no segment ab in S has p in its interior.
- The boundary of $CH(Z)$ is a simple polygon with a subset of Z as its corners.
- $CH(Z)$ is considered determined once its corners ordered around the boundary are found.
- Simplifying assumptions: No pair of Z-points has the same x- or y-coordinate and/or no three points colinear.

Equivalent Formulations

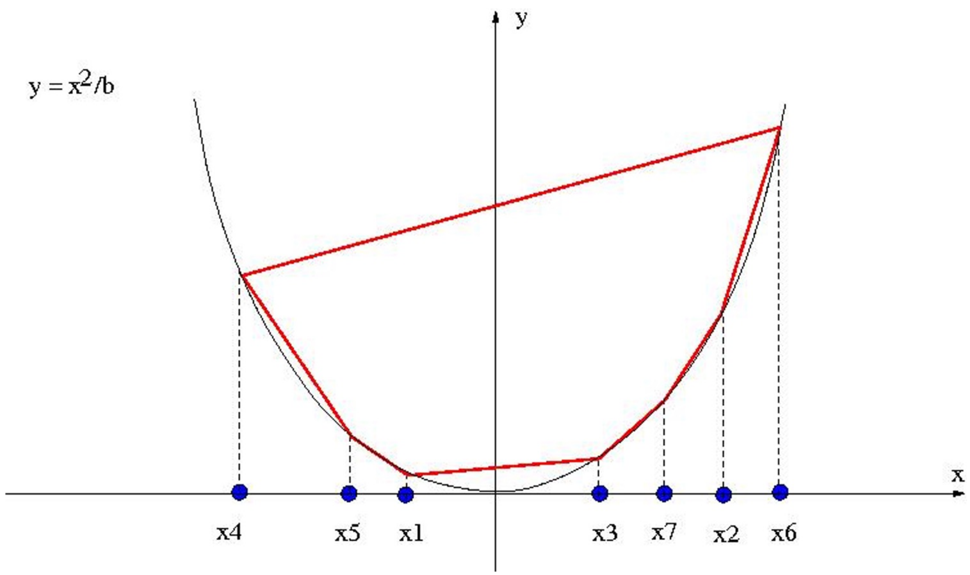
- Smallest possible convex set containing *Z*.
- Intersection of all convex sets containing *Z*.
- Intersection of all half-planes containing *Z*.
- Set of all convex combinations of points in *Z*

$$\sum_{i=1}^{n} a_i z_i, \sum_{i=1}^{n} a_i = 1, a_i \geq 0$$

- Set of convex combinations of *d*+1 points in *Z*.
- Convex set with the smallest possible perimeter and containing *Z*.

Lower Bound

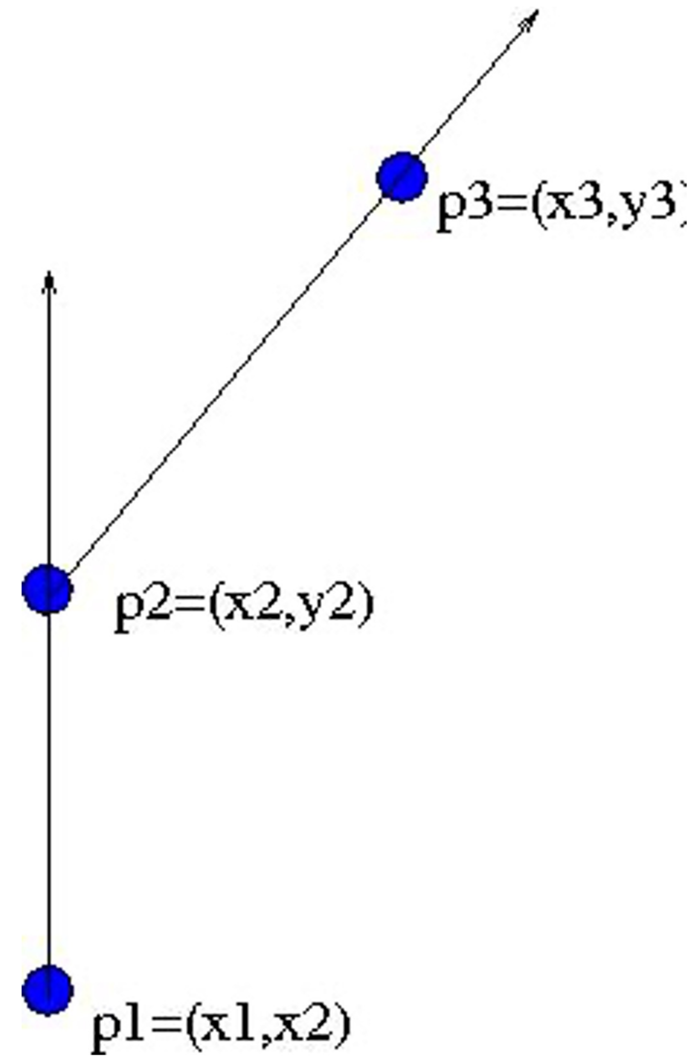
- Sorting can be transformed in $O(n)$ time into the convex hull problem.
- Sorting: $\Omega(n \log n)$
- Convex hull: $\Omega(n \log n)$
- Transformation:



$$x_i \rightarrow (x_i, x_i^2/b)$$

Left and Right Turns

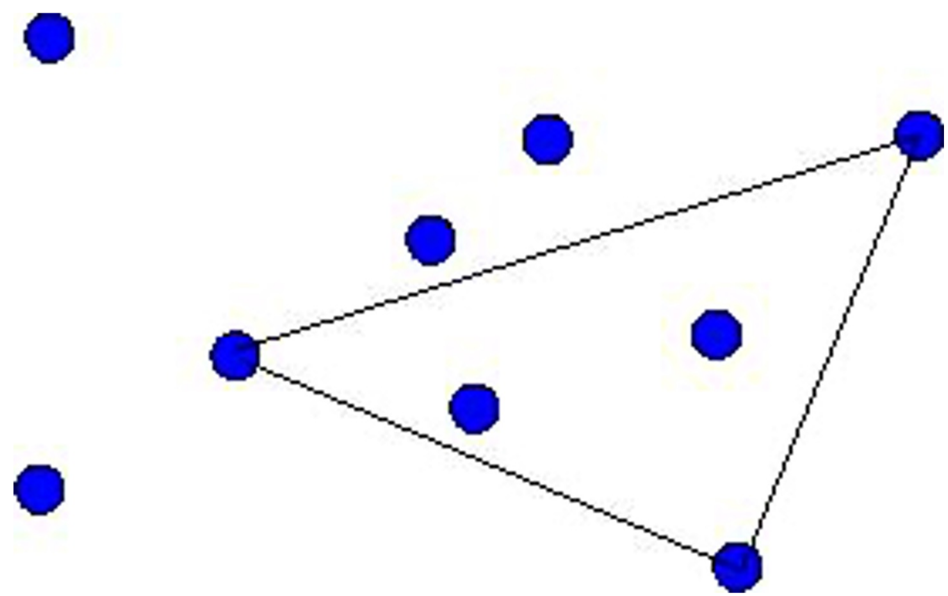
- Points p_1, p_2, p_3 make a right turn at p_2 iff p_3 is to the right or on the line through p_1 and p_2.
- Otherwise p_1, p_2, p_3 make a left turn at p_2.



$$\begin{Vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{Vmatrix} > 0 \text{ - left turn}$$

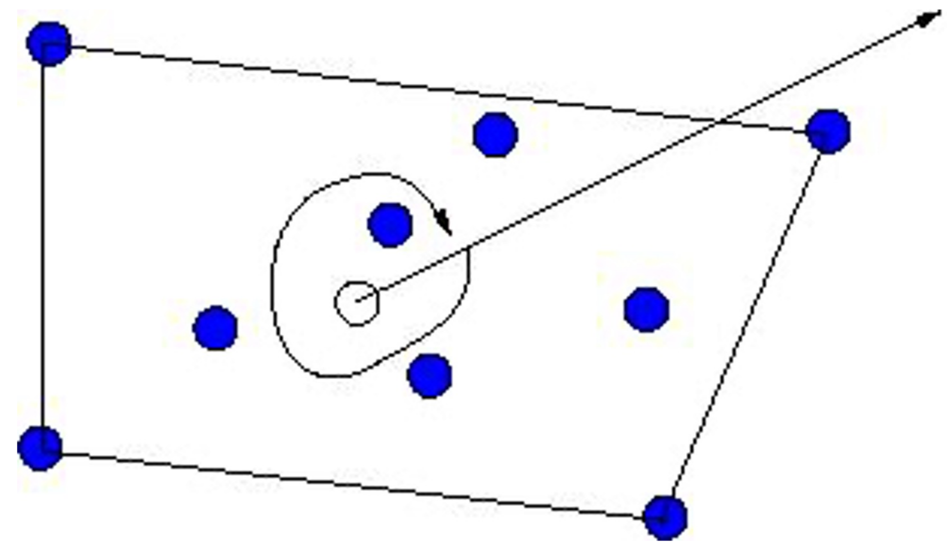
Point Pruning

- A Z-point which is not a corner is inside a triangle on Z.
- $O(n^3)$ triangles. $O(n)$ inclusion test for each triangle.
- $O(n^4)$ in total.
- Ordering remains to be determined.



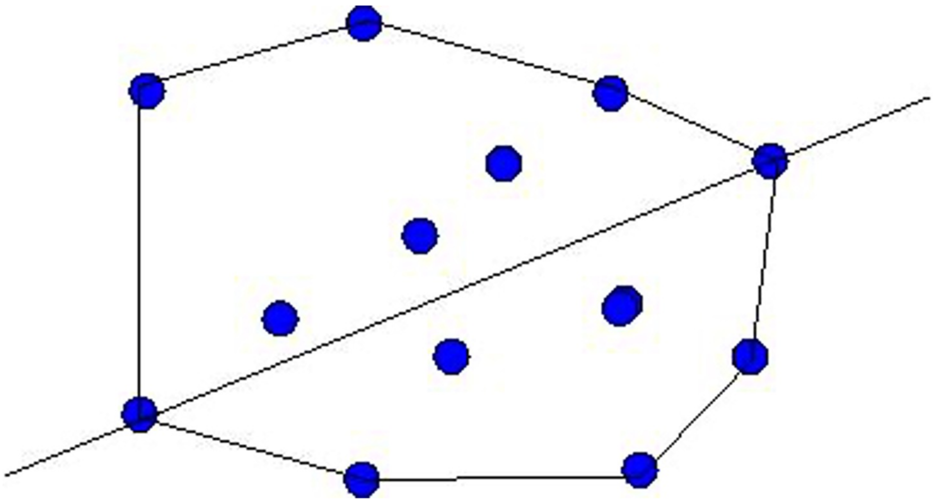
Point Pruning - Ordering

- Take any point in *CH*(*Z*).
- Sort corners around it.
- How do we find a point in *CH*(*Z*)? Do not think too hard. It is trivial :-)



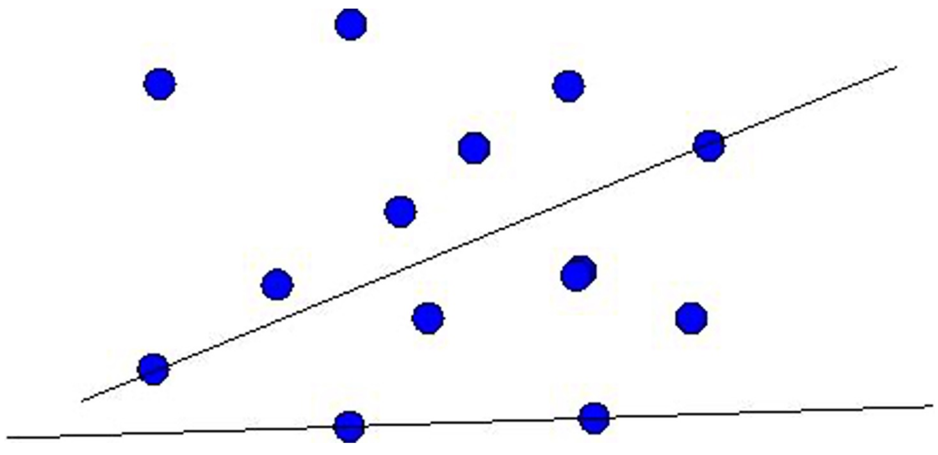
Point Pruning – Ordering Without Polar Coordinates

- Draw line through the leftmost and rightmost corners.
- Sort corners above the line by increasing *x*-coordinate.
- Sort corners below the line by decreasing *x*-coordinate.
- All this can be done in $O(n \log n)$ time.



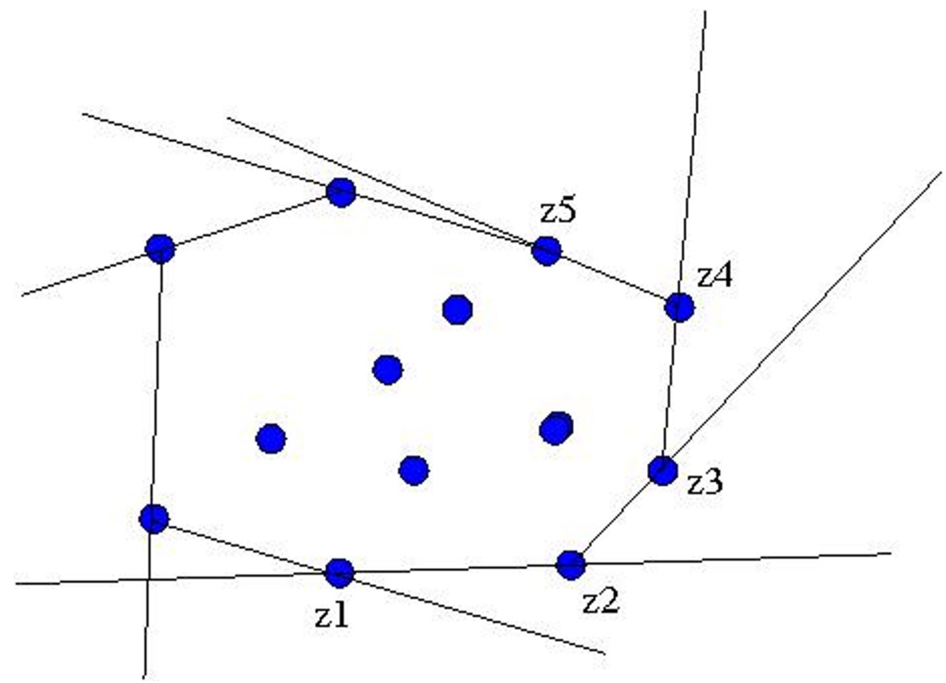
Edge Pruning

- Identify boundary edges rather than corners.
- Consider a line through a pair of *Z*-points.
- Some of these lines are more interesting than others.
- $O(n^3)$ algorithm.
- Sorting of corners remains.



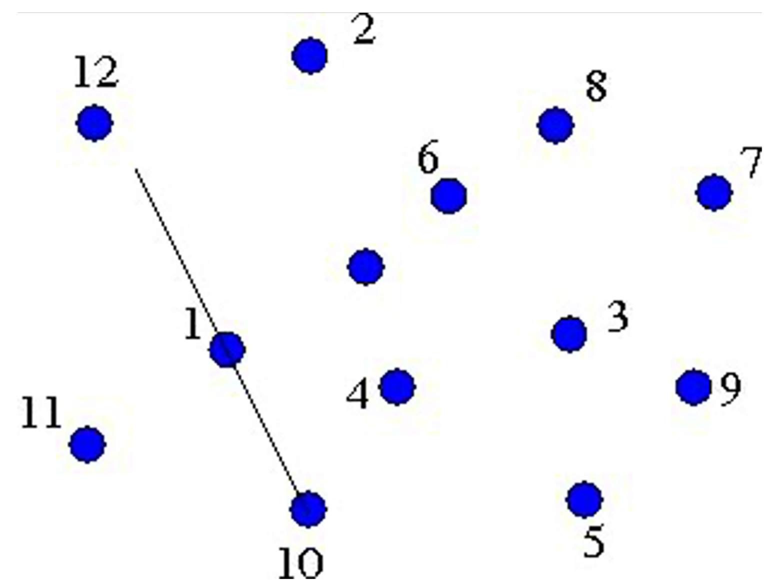
Jarvis' March (1973)

- Improvement of the edge pruning algorithm.
- When a boundary edge $z_i z_j$ has been identified, there must exist another boundary edge with z_j as one of its endpoints.
- How do we find the first boundary edge?
- Corner sorting not needed.



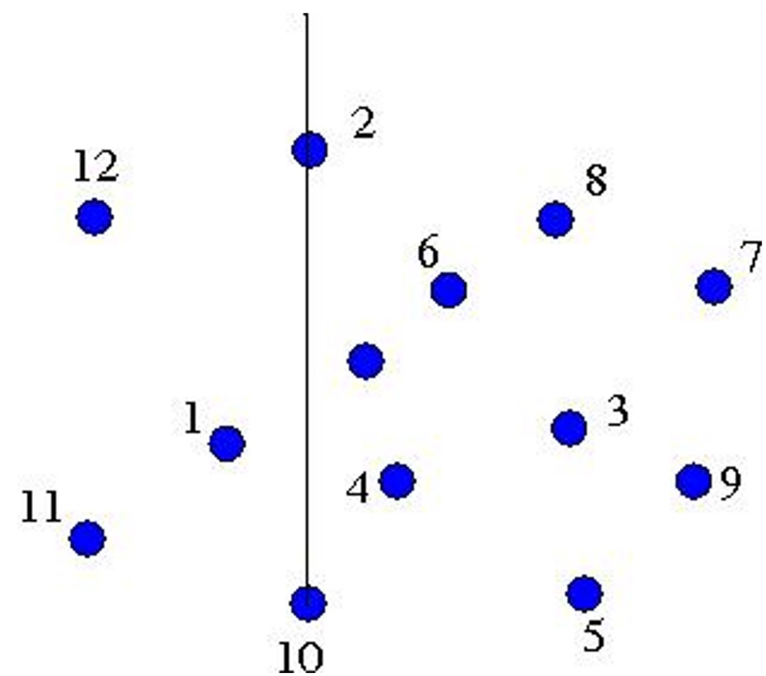
Implementation of Jarvis' March

- Use left and right turns to find next boundary edge in $O(n)$ time.
- Jarvis march takes $O(hn)$ time where h is the number of corners (originally unknown).
- Expected number of corners of points independently and uniformly distributed within a unit circle is $O(n^{1/3})$.
- Can be generalized to higher dimensions.



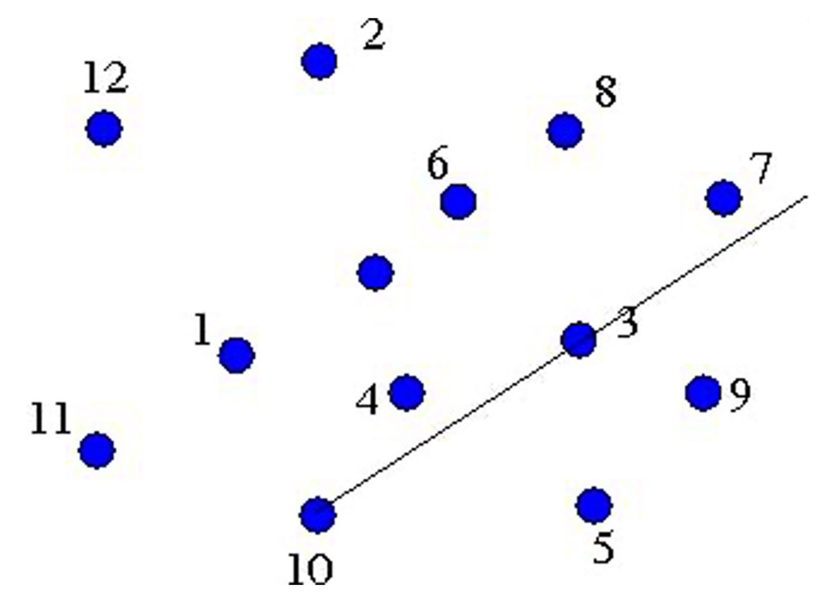
Implementation of Jarvis' March

- Use left and right turns to find next boundary edge in $O(n)$ time.
- Jarvis march takes $O(hn)$ time where h is the number of corners (origianlly unknown).
- Expected numbers of corners of points independently and unoformly distributed within a unit circle is $O(n^{1/3})$.
- Can be generalized to higher dimensions.



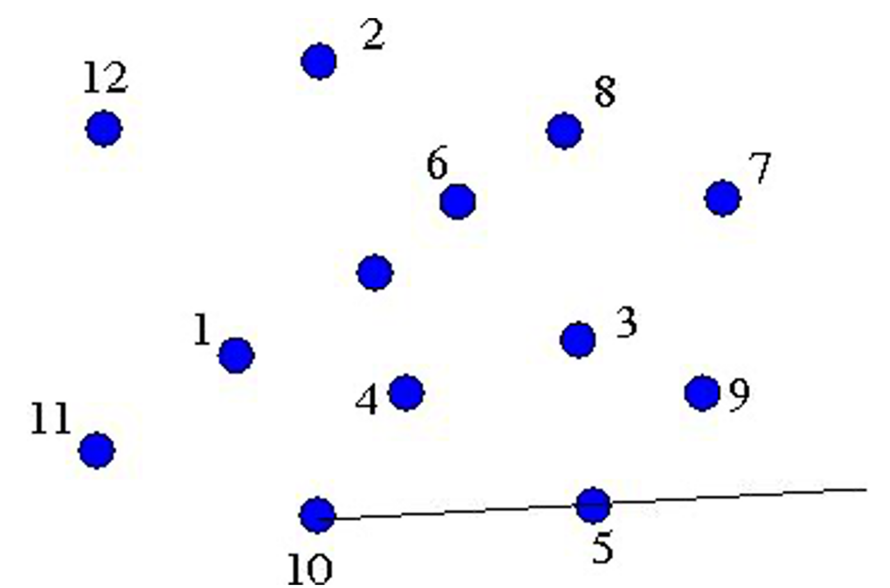
Implementation of Jarvis' March

- Use left and right turns to find next boundary edge in $O(n)$ time.
- Jarvis march takes $O(hn)$ time where h is the number of corners (origianlly unknown).
- Expected numbers of corners of points independently and unoformly distributed within a unit circle is $O(n^{1/3})$.
- Can be generalized to higher dimensions.



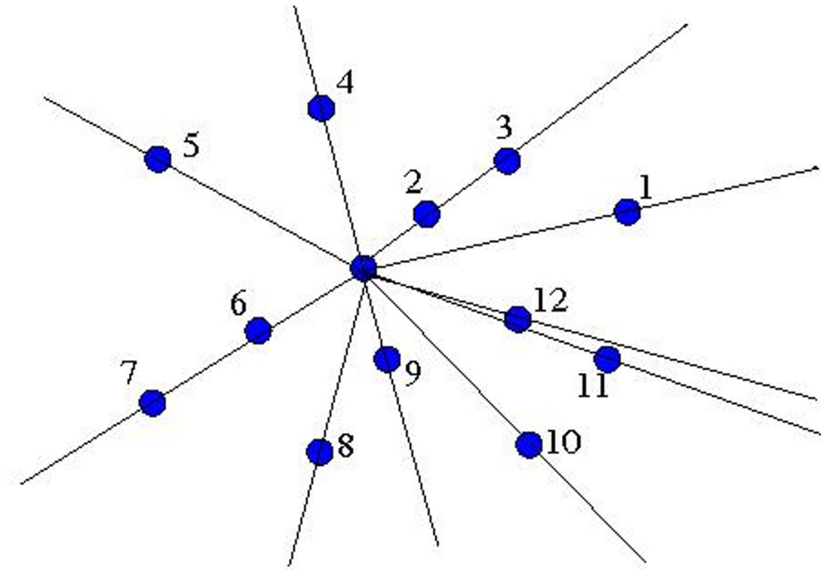
Implementation of Jarvis' March

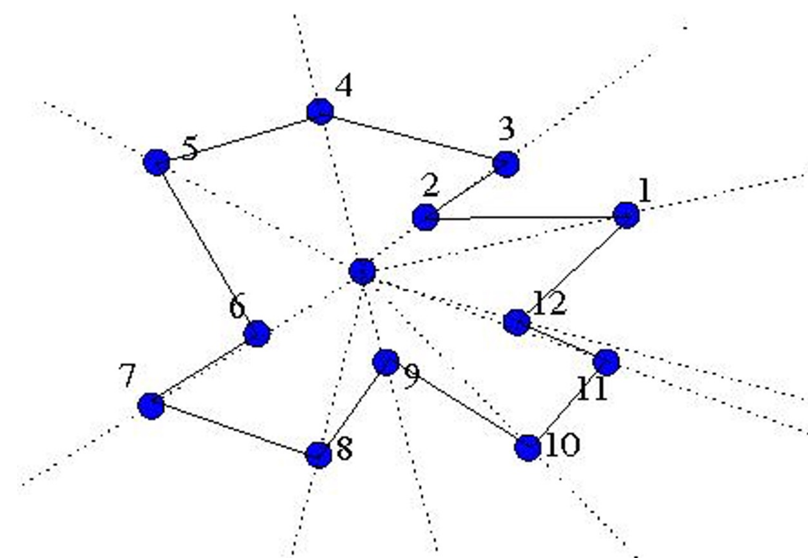
- Use left and right turns to find next boundary edge in $O(n)$ time.
- Jarvis march takes $O(hn)$ time where h is the number of corners (origianlly unknown).
- Expected numbers of corners of points independently and unoformly distributed within a unit circle is $O(n^{1/3})$.
- Can be generalized to higher dimensions.



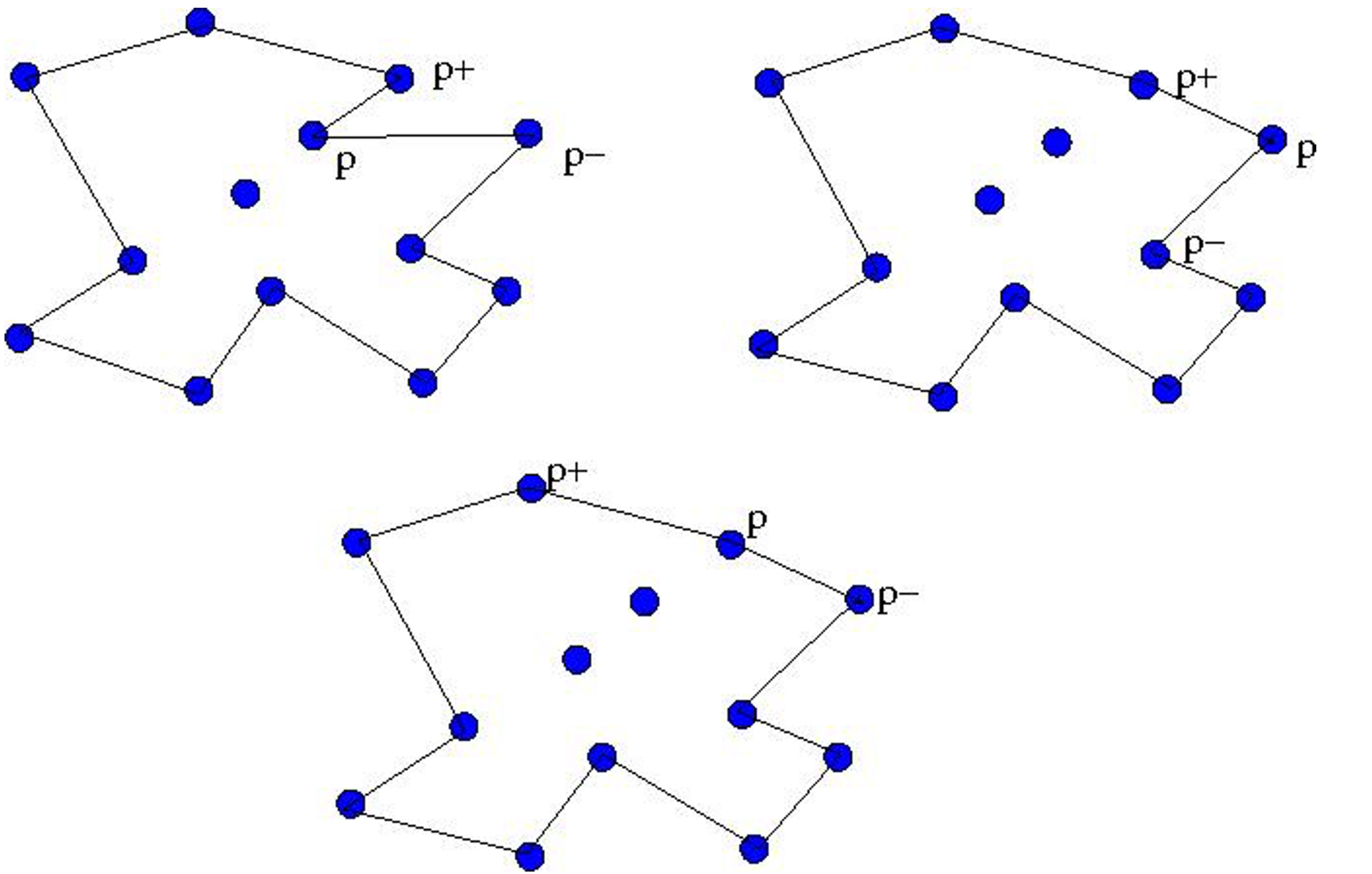
Graham's Scan (1972)

- Sort points around some point of *CH*(*Z*). Points at the same angle are sorted by their distance.
- Construct the polygon defined by the sorting.





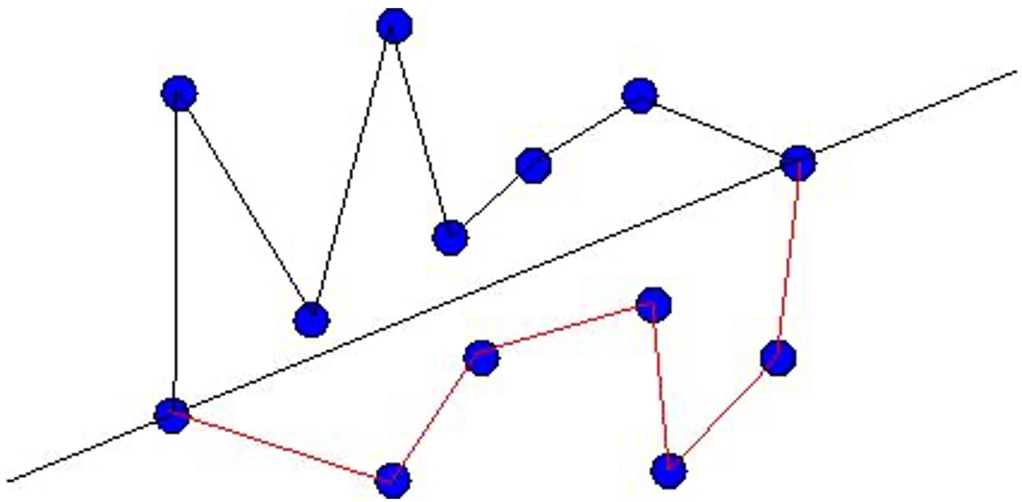
Graham's Scan Continued



Correctness of Graham's Scan

- Will never go backwards behind the initial corner.
- Invariant: When arriving at some point *p*, all points between the initial corner and *p* are left turns on the polygonal line constructed so far.
- After arriving at the initial corner by a forward step, we get a polygon where all corners are left turns.

Graham's Scan without Polar Coordinates



Complexity of Graham's Scan

- # of backward steps is $O(n)$: one point is removed at every backward step.
- # of forward steps is $O(n)$:
  - first forward step from a given point can be made only once.
  - every subsequent forward step from the same point occurs right after a backward step.
- Both forward and backward steps require $O(1)$ time.
- Graham's scan requires $O(n)$ time + $O(n \log n)$ sorting.