http://www.personal.kent.edu/~rmuhamma/Maingif/redline.gif

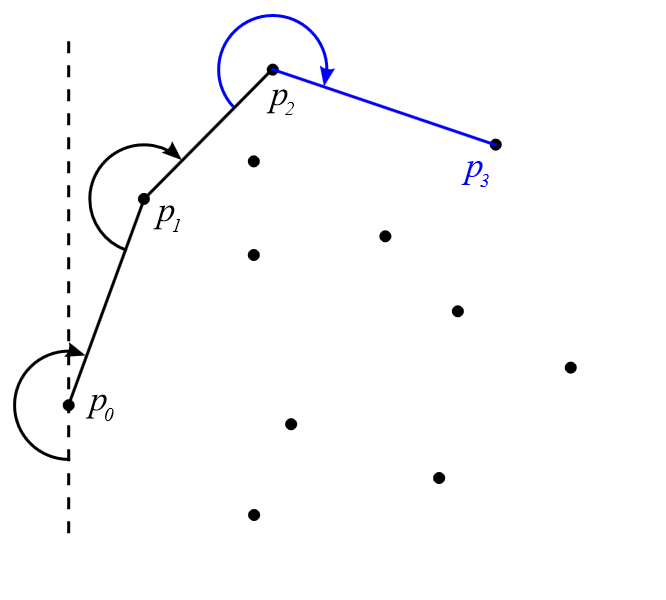
**Jarvis March**

http://www.personal.kent.edu/~rmuhamma/Maingif/redline.gif

Jarvis march computes the *CH*(*Q*) by a technique known as gift wrapping or package wrapping.

**Algorithm Jarvis March**

* First, a base point *po* is selected, this is the point with the minimum y-coordinate.
  + Select leftmost point in case of tie.
* The next convex hull vertices *p*1 has the least polar angle w.r.t. the positive horizontal ray from *po*.
  + Measure in counterclockwise direction.
  + If tie, choose the farthest such point.
* Vertices *p*2, *p*3, . . . , *pk* are picked similarly until yk = ymax
  + *pi*+1 has least polar angle w.r.t. positive ray from *po*.
  + If tie, choose the farthest such point.



* The sequence *po*, *p*1, . . . , *pk* is right chain of *CH*(*Q*).
* To choose the left chain of *CH*(*Q*) start with *pk*.
  + Choose pk+1 as the point with least polar angle w.r.t. the negative ray from *pk*.
  + Again measure counterclockwise direction.
  + If tie occurs, choose the farthest such point.
  + Continue picking *pk*+1, *pk*+2, . . . , *pt* in same fashion until obtain *pt* = *po*.

**Complexity of Jarvis March**

For each vertex p belongs to *CH*(*Q*), it takes

* O(1) time to compare polar angles.
* O(n) time to find minimum polar angel.
* O(n) total time.

If *CH*(*Q*) has h vertices, then running time O(*nh*). If h = *o*(*lg n*) (this is little Oh), then this algorithm is asymptotically faster than the Graham’s scan. If points in set Q are generated by random generator, the we expect *h* = *c lg n* for *c*≈1.

In practice, Jarvis march is normally faster than Graham’s scan on most application. Worst case occurs when O(n) points lie on the convex hull i.e., all points lie on the circle.

http://www.personal.kent.edu/~rmuhamma/Maingif/redline.gif