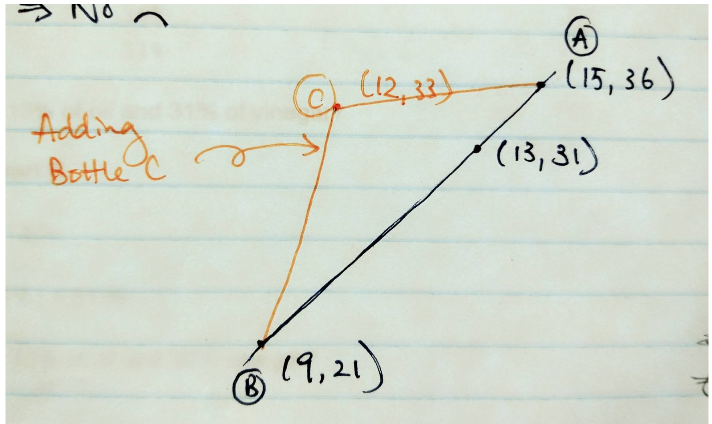
CPSC 420 - Jan 4 & 6 Lecture Notes

Naomi Kam



EXAMPLE: Salad Dressing

Bottle	Oil	Vinegar
A	15%	36%
B	9%	21%

- Can we mix bottles to get 13% of oil and 31% of vinegar?

  -> Yes! 2 Parts A and 1 part B

  Oil: $\frac{2}{3}(15\%) + \frac{1}{3}(9\%) = 13\%$

  Vinegar: $\frac{2}{3}(36\%) + \frac{1}{3}(21\%) = 31\%$

- Can we mix bottles to get 12% of oil and 30% vinegar?

  -> No :(

  Referring to the figure above, the black line segment reflects the solutions obtainable by mixing bottles A and B.

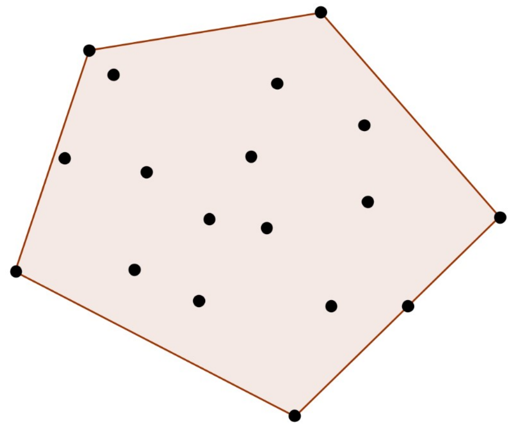
  If another bottle C of 12% oil and 33% vinegar is added (as shown by the orange line segment), then we can not only obtain a mixture of 12% oil and 30% vinegar, but any points enclosed within the triangle formed by A, B, and C.

A **mixture** is a **convex combination** of points representing the contents of the bottles, where points $p_1, p_2, \cdots, p_n$ is $\sum_{i=1}^{n} \alpha_i p_i$ such that

- $\sum_{i=1}^{n} \alpha_i = 1$
- $\alpha_i \geq 0$ for all i

Convex Hulls (CH(P))

Convex hull (CH(P)) of a set of points P is the smallest convex set containings P, the set of all convex combinations of P.



A set T is **convex** if for all $a, b \epsilon T$, segment ab is in T.

The boundary of the convex combination are bounded by the perimeter points of the set.

Problem

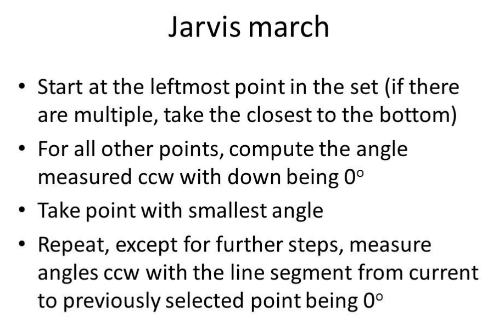
Input: set of points $P = p_1 p_2 \cdots p_n$

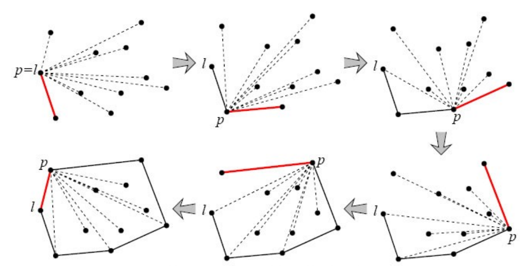
Output: convex hull of P

Image borrowed from https://hcmop.wordpress.com/tag/combinatorial-geometry/

Approach 1: Jarvis March ("Gift-wrapping") 1973

Gist of Algorithm





The execution of Jarvis's March.

Psuedocode from Class

1. Find point p_0 in P with minimum y-coordinate ---- O(n)
2. set $h = 0$ ---- O(1)
3. **repeat** for each $p \epsilon P$

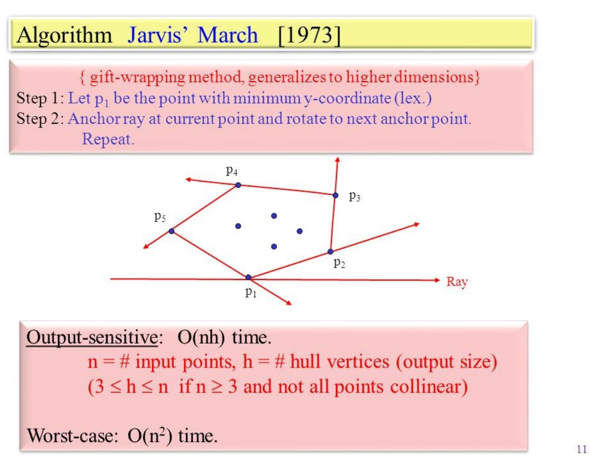
   if rightturn(p_h, q, p)

   $q = p$

   $p_{h+1} = q$

   $h = h + 1$

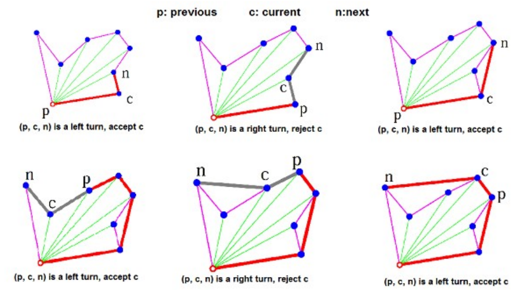
   until $p_h = p_0$



Slide borrowed from http://slideplayer.com/slide/3893034/ , http://www.tcs.fudan.edu.cn/rudolf/Courses/Algorithms/Alg_cs_07w/Webprojects/Zhaobo_hull/ and http://slideplayer.com/slide/2411187/

Approach 2: Graham's Scan (1972)

```
GRAHAM-SCAN(Q)
let p0 be the point in Q with the minimum y-coordinate,
      or the leftmost such point in case of a tie
let ⟨p1, p2, ..., pm⟩ be the remaining points in Q,
      sorted by polar angle in counterclockwise order around p0
      (if more than one point has the same angle, remove all but
      the one that is farthest from p0)
let S be an empty stack
PUSH(p0, S)
PUSH(p1, S)
PUSH(p2, S)
for i = 3 to m
      while the angle formed by points NEXT-TO-TOP(S), TOP(S),
                and pi makes a nonleft turn
            POP(S)
      PUSH(pi, S)
return S
```



In the above algorithm and below code, a stack of points is used to store convex hull points. With reference to the code, p is next-to-top in stack, c is top of stack and n is points[i].

Psuedocode borrowed from http://stackoverflow.com/questions/11466802/convex-hull-sorting-step and http://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/