Introduction to Algorithms

Mong-Jen Kao (高孟駿)

Tuesday 10:10 – 12:00

Thursday 15:30 – 16:20

Program Assignment - I

General Tips

- Use C++ STL ***basic data container*** and ***basic algorithms*** to compose your solution.
  - pair, ***vector***, ***array***, etc.
  - ***sort*** algorithm with ***custom compare function***.
- The loading for program assignment becomes *reasonable* only when you use these tools properly.
- If unsure, refer to the sample codes and look up the references for C++ STL.

Program Assignment - I

Cross-product of 2D Vectors

- Let $\vec{u}, \vec{v} \in \mathbb{R}^3$ be two vectors lying in the $x - y$ plane.

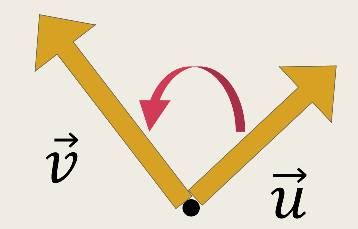
$$\vec{u} = (\mathrm{a}_1, \mathrm{b}_1, 0), \qquad \vec{v} = (a_2, b_2, 0).$$

- Then $\vec{u} \times \vec{v}$ will be parallel to the z-axis, i.e.,

$$\vec{u} \times \vec{v} = (0, 0, k), \qquad \text{where } k = a_1 b_2 - a_2 b_1.$$

- Furthermore,

  the value of k will satisfy the following.



- Let $\vec{u}, \vec{v} \in \mathbb{R}^3$ be two vectors lying in the $x - y$ plane.

$$\vec{u} = (\mathrm{a}_1, \mathrm{b}_1, 0), \qquad \vec{v} = (a_2, b_2, 0).$$

- Then $\vec{u} \times \vec{v}$ will be parallel to the z-axis, i.e.,

$$\vec{u} \times \vec{v} = (0, 0, k), \qquad \text{where } k = a_1 b_2 - a_2 b_1.$$

- Furthermore,

  the value of k will satisfy the following.

  – If $\vec{u}$ to $\vec{v}$ is a ***counter-clockwise*** rotation, then $\boldsymbol{k > 0}$.

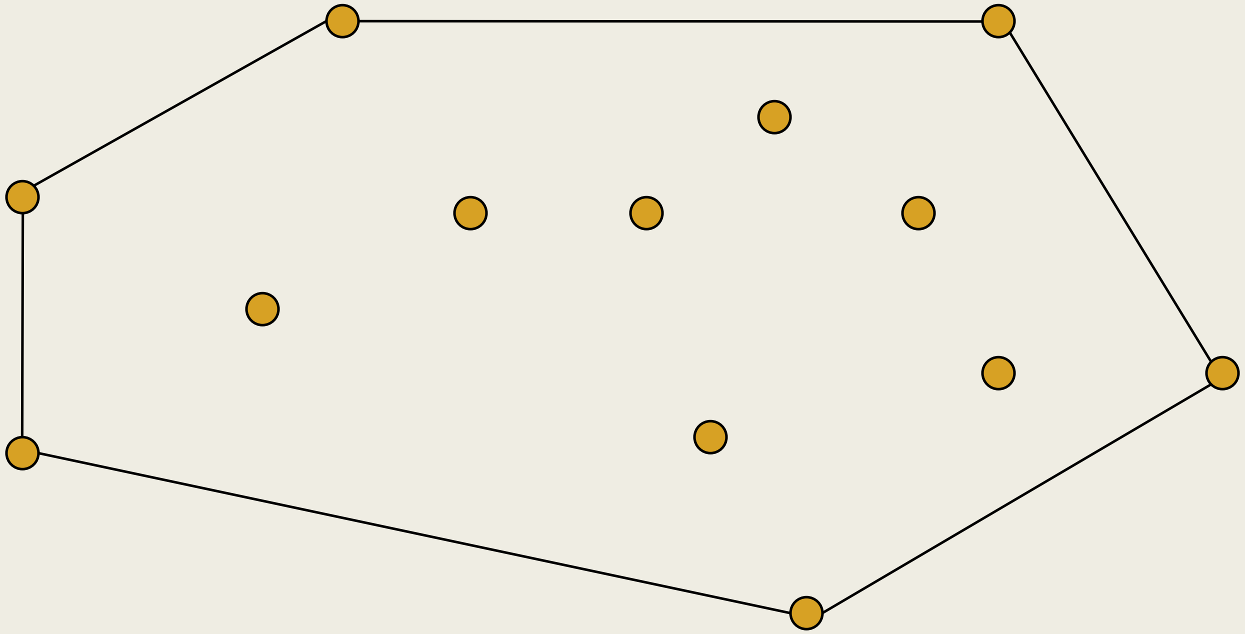
  – If $\vec{u}$ to $\vec{v}$ is a ***clockwise*** rotation, then $\boldsymbol{k < 0}$.

  – If $\vec{u}$ and $\vec{v}$ are ***parallel***, then $\boldsymbol{k = 0}$.

The Convex Hull Problem

- Given a set P of points in the plane,

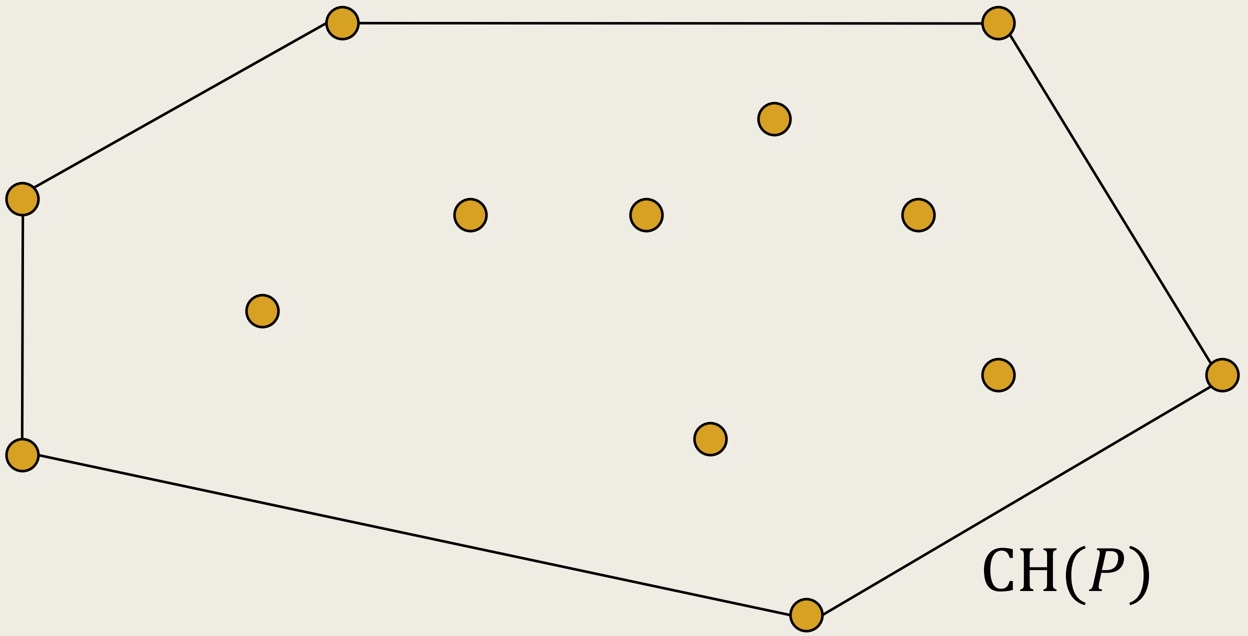
  the **Convex Hull** of P is the smallest **convex polygon** that contains P.



The Convex Hull Problem

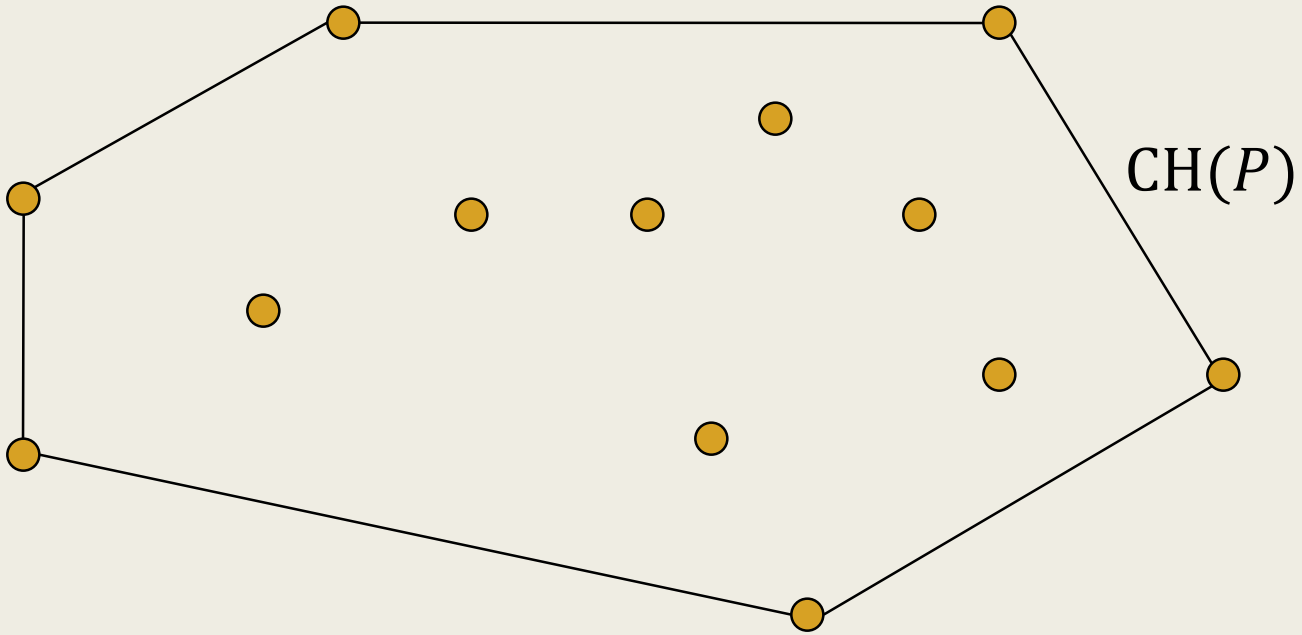
- Given a set P of points in the plane,

  the **Convex Hull** of P is the smallest convex polygon that contains P.



- Given a set P of points in the plane,

  the **Convex Hull** of P is the smallest convex polygon that contains P.



A characterization of the hull vertices.

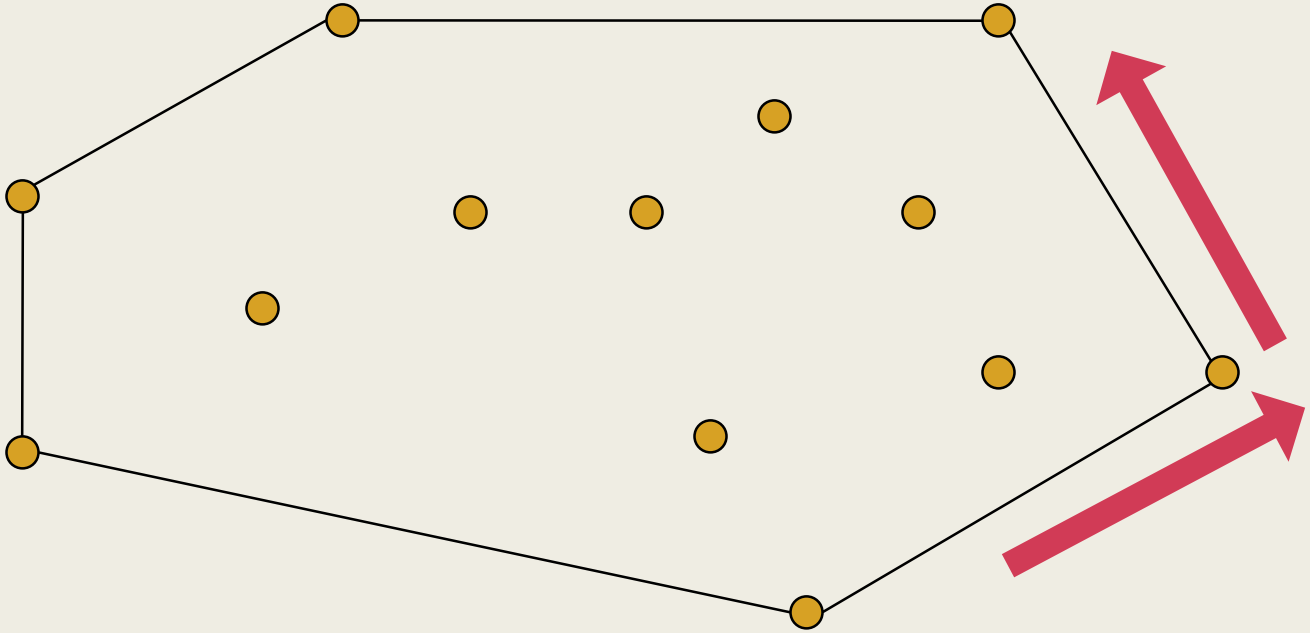
Definition. A point v ∈ P is a ***vertex*** of Convex Hull of P *if* there exists no $q, r \in \mathrm{CH}(P)$ such that $v = (q + r)/2$.

The Convex Hull Problem

- The Convex Hull can be constructed in O(n log n) time.
  - There are many ways to do this.
  - In the following, we will see the **Graham Scan method**, which requires only sorting and products of the vectors.

Observation

- Imagine that you are walking along the boundary of the convex hull in counter-clockwise order.
  - Then, you always make ***left-turns*** at the vertices.

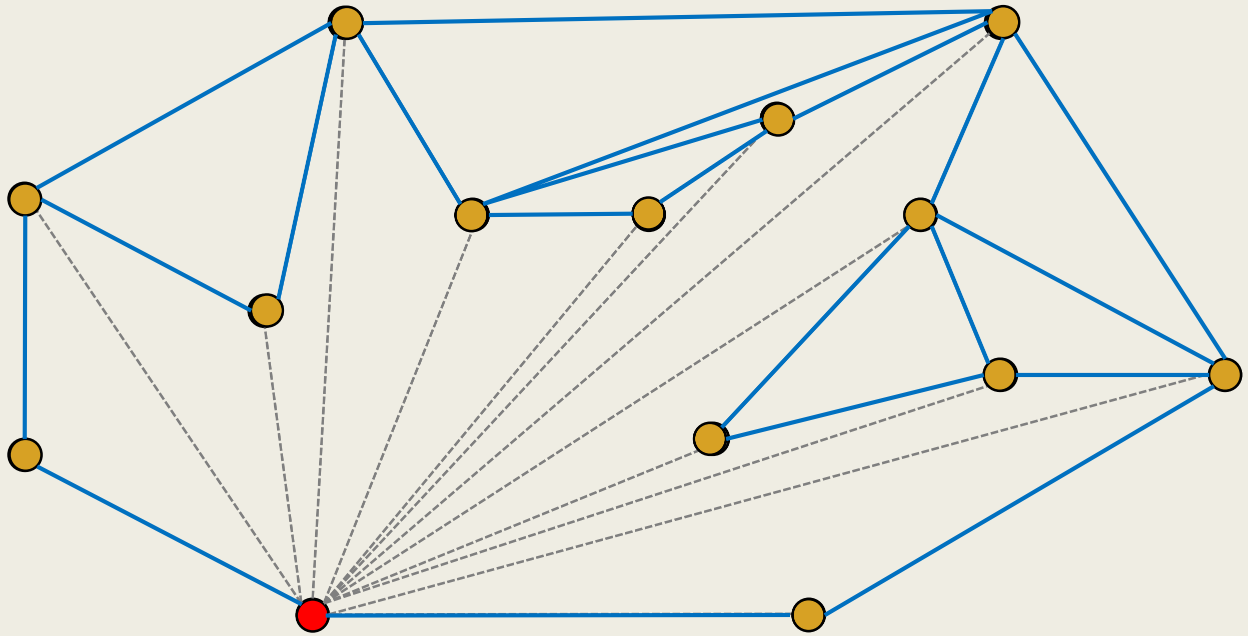


The Graham Scan Algorithm

1. First, pick the point with smallest y-coordinates.
   If ties happen, further pick the vertex with the smallest x-coordinate.
   Let this point be p_1.
   - p_1 must be one of the vertices of the convex hull.

2. Sort the remaining vertices according to the vectors formed from p_1 to them in counter-clockwise order. Break ties according to their lengths.
   - Use cross-product (for counter-clockwise order) and
     inner-product (for the length) to do the comparison.

- Traverse the points in sorted order to form the hull boundary.
  - During the process, make sure that **the last three vertices always form a left-turn**.
  - If not, the second last vertex is deleted until the above is true or only two vertices are left.



The Graham Scan Algorithm

3. Let L = { p_1 } be the initial list of hull vertices.

4. Traverse the remaining points in sorted order.
   - Add the current point to the end of L.
   - While $|L| \geq 3$ and the last three points of L does not form a left-turn
     - Delete the second-last point from L

5. Output L.

This process takes $O(n)$ time.

The overall complexity is $O(n \log n)$.