COL 758: Advanced Algorithms

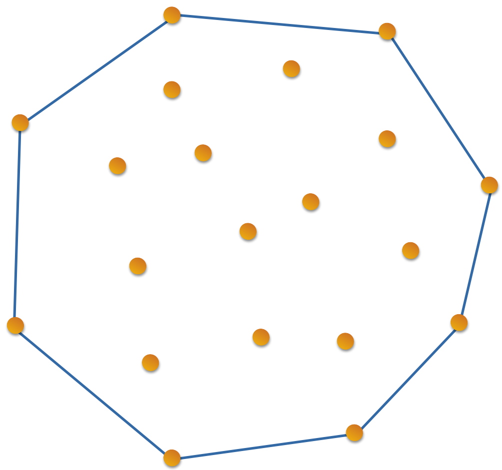
Lecture 12-13

Randomized Incremental Construction

CONVEX HULL Problem

Given: Set S of n points in 2D plane.

Output: Convex polygon of smallest size enclosing all points in S.



Randomized Approach:
Permute the points randomly, and add them one by one.

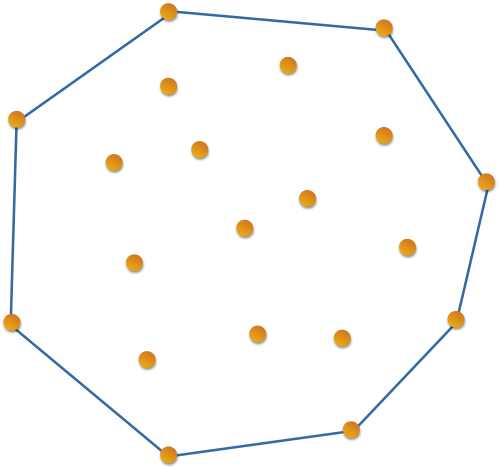
Time: $O(n \log n)$

Assumption: No three points are collinear.

CONVEX HULL Problem

Given: Set S of n points in 2D plane.

Output: Convex polygon of smallest size enclosing all points in S.



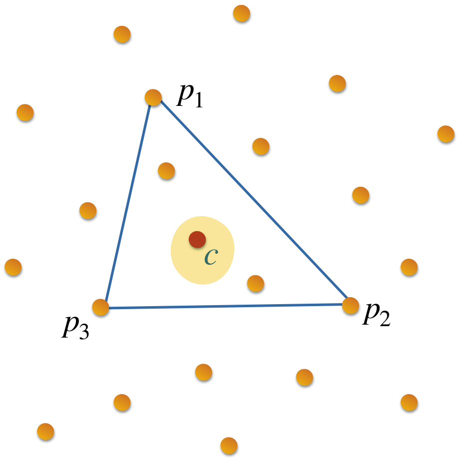
Notations:

- S : Input set of n points
- $(p_1, p_2, \ldots, p_n)$: Random permutation of points in S
- S_i : Set of first i points, i.e. $(p_1, \ldots, p_i)$
- c : Some fixed point in the convex hull.
  (c maynot be in input set S.)

Randomized Algorithm

Given: Set S of n points in 2D plane.

Output: Convex polygon of smallest size enclosing all points in S.

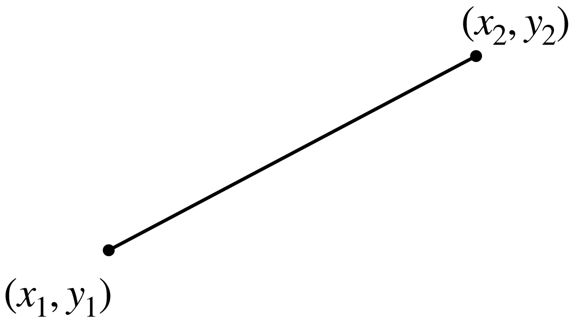


1. Let $P = (p_1, \ldots, p_n)$ be a random permutation of the n points.
2. Initialize **CH** to be Triangle with p_1, p_2, p_3 as end-points.
3. Let c = centroid of Triangle with p_1, p_2, p_3 as end-points.
4. For $(i = 4$ to $n)$:

   If $(p_i \notin$ **CH**)**:** Insert p_i and update convex-hull **CH**.

Remark: Throughout the algorithm run, point c is **fixed** and **lies inside CH**

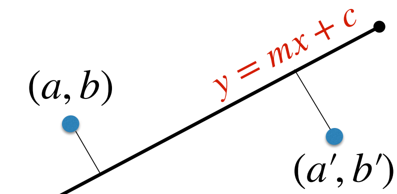
Geometric Tools



Find equation $(y = mx + c)$ of line passing through **two points**

We can find m, c in $O(1)$ time by solving

$$y_1 = mx_1 + c$$
$$y_2 = mx_2 + c$$

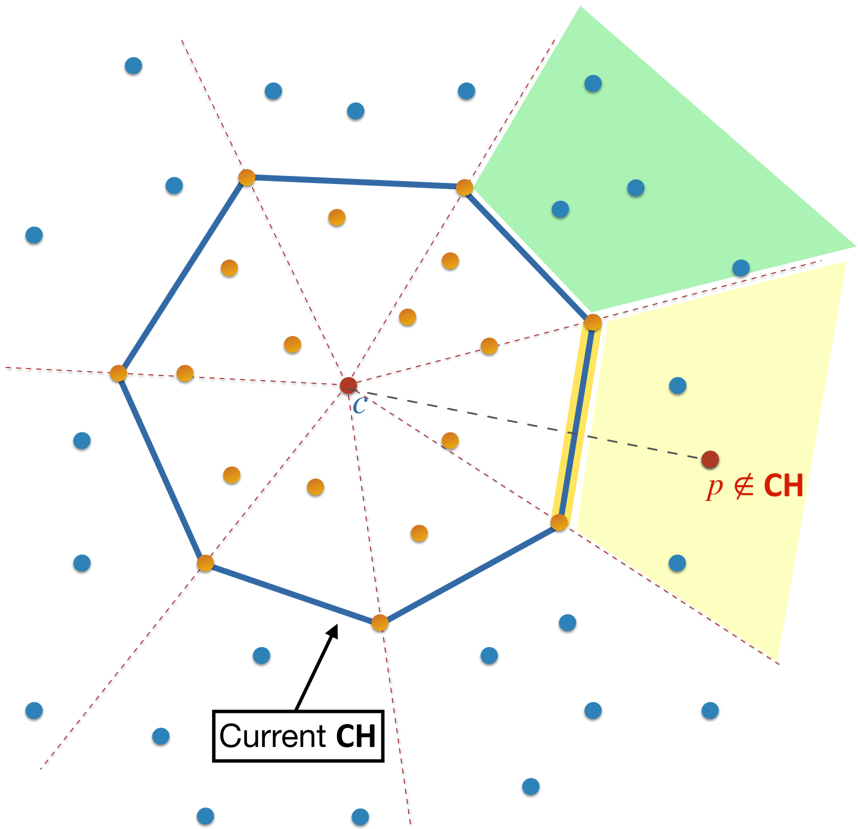


We can check wether two points are in same side or different side of a **line** in $O(1)$ time as follows

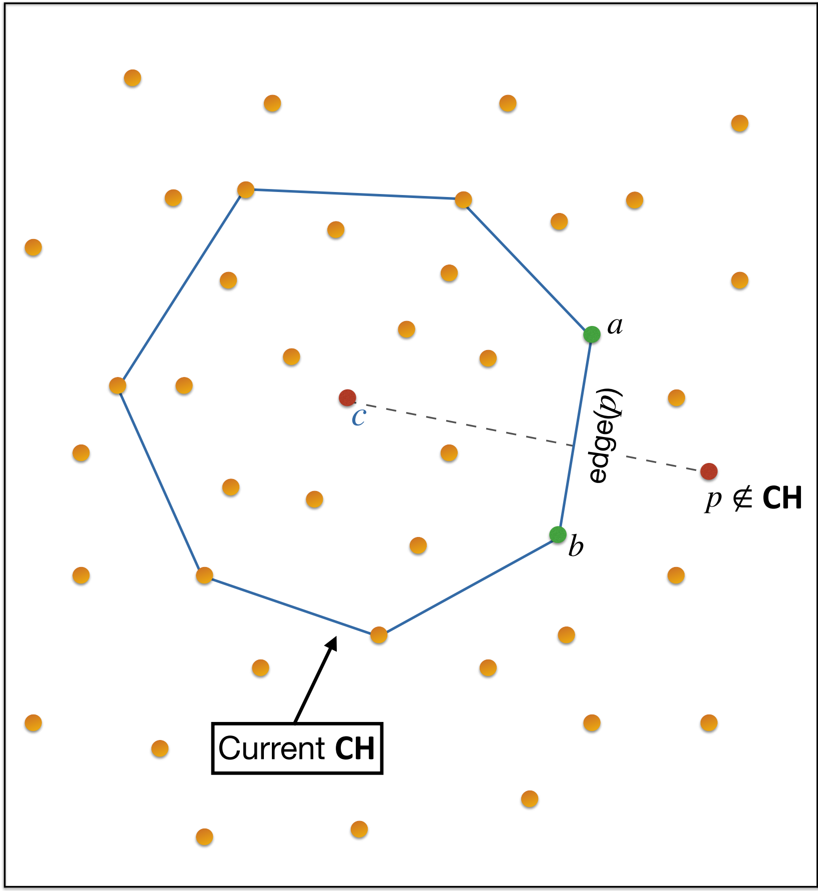
Substitute (a, b) and (a', b') in eq. $(y - mx - c)$

- If signs are same $\Rightarrow$ Same side of line
- If signs are different $\Rightarrow$ Different side of line

Another Notation: Edge of a point p lying outside the CH



Question: How many times can edge of a point p change?



Observation 1:

$$Prob\left(\text{edge}(p) \text{ is updated in } \mathbf{CH}(S_i) \;\middle|\; S_i,\, p \notin S_i\right) = \frac{2}{i}$$

Proof:

Suppose $(a, b) = \text{edge}(p)$ in the convex hull $\mathbf{CH}(S_i)$.

Then,

(a, b) is also edge(p) is $\mathbf{CH}(S_{i-1})$

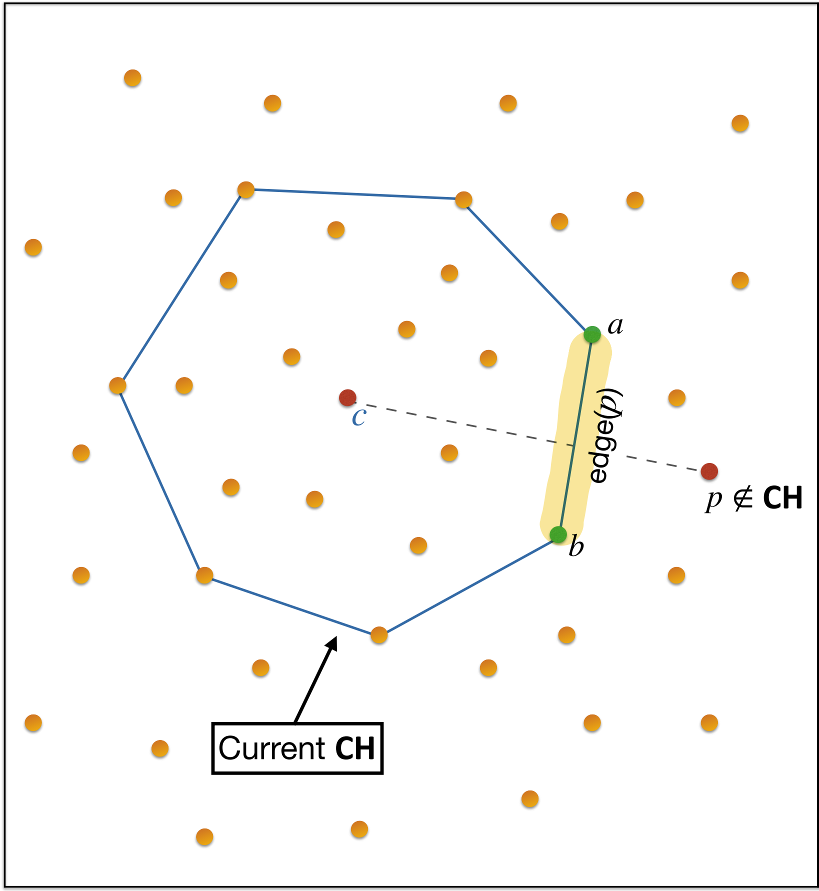
iff

a, b are not point p_i in set S_i.

Now,

$$Prob\left(a \neq p_i \text{ and } b \neq p_i \;\middle|\; S_i,\, p \notin S_i\right) = \frac{2}{i}$$

Question: How many times can edge of a point p change?



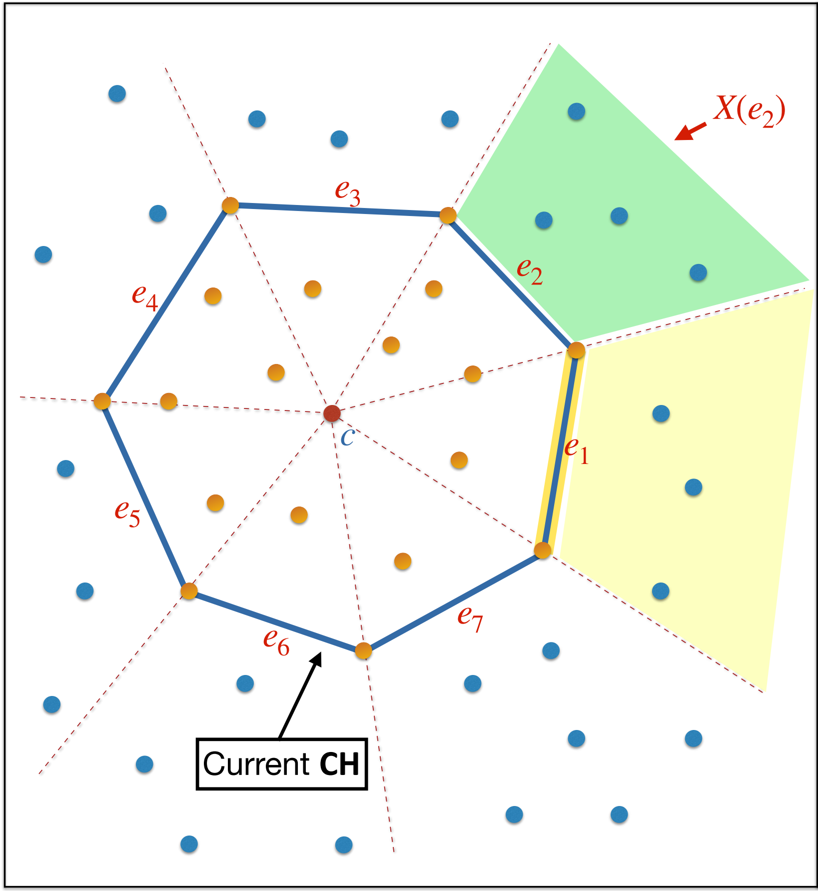
Observation 1:

$$Prob\left(\text{edge}(p) \text{ is updated in } \mathbf{CH}(S_i) \;\middle|\; S_i,\, p \notin S_i\right) = \frac{2}{i}$$

Observation 2:

The expected number of times edge(p) is updated throughout the algorithm run, for a fixed point p is $O\left(\sum_{1 \le i \le n} 2/i\right) = O(\log n)$.

Data-Structures



1. Each point p stores:

   — flag(p) $= \begin{cases} 1 & \text{if } p \notin CH \\ 0 & \text{otherwise} \end{cases}$

   — edge(p), if $p \notin$ **CH**

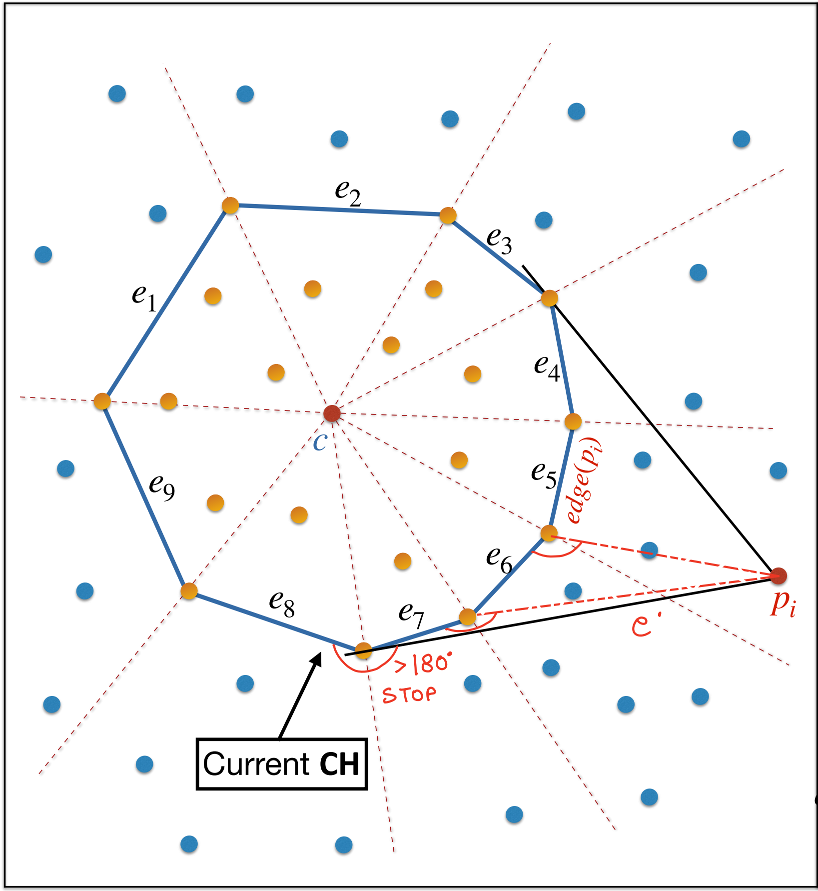
2. Store the edges of **CH** in circular Link-List.

e_1	e_2	e_3	e_4	e_5	e_6	e_7

3. Each edge e stores

$$X(e) = \left\{ p \in S \setminus \mathbf{CH} \;\middle|\; e = \text{edge}(p) \right\}$$

Insertion of Point p_i

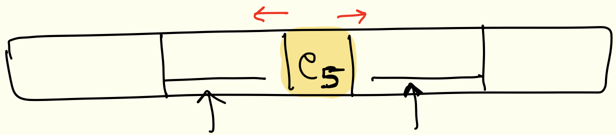


Ques: How to find the set Y of all edges that will be destroyed?

In figure $Y = \{e_4\, e_5\, e_6\, e_7\}$.

First step is to extract edge(p_i).

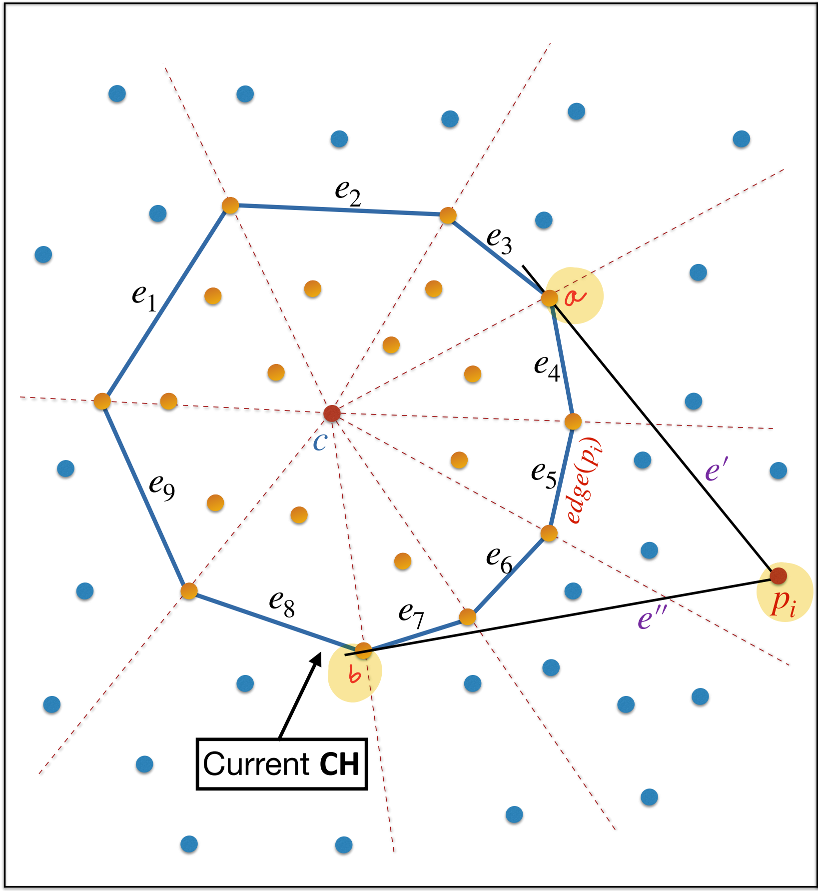
Next we move from edge(p_i) in Doubly Link list in both directions, to find Y.



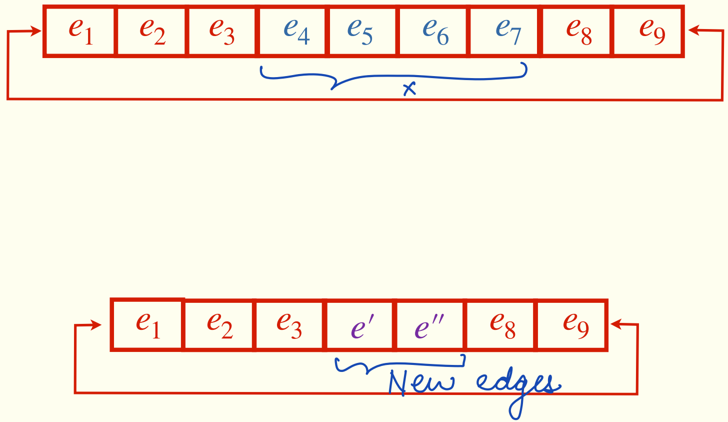
Approach 1 : Include all those edges for which p_i and c are on opposite side

Approach 2: Stop if angle from p_i to edge becomes larger than 180°

Insertion of Point p_i



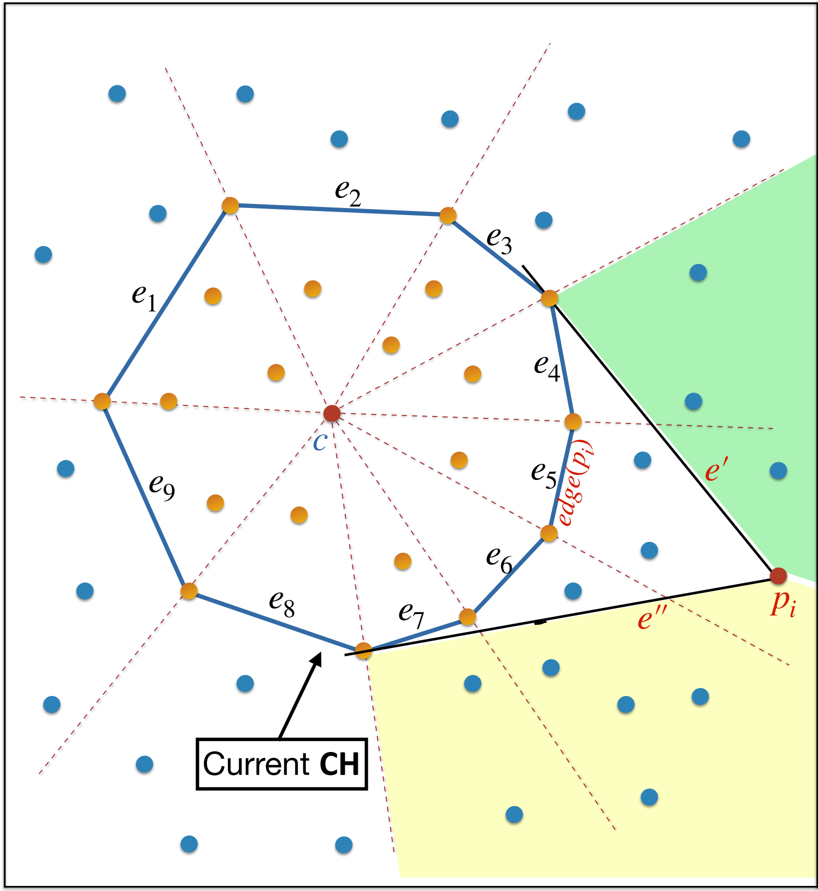
Circular link list of edges after insertion of p_i



≥ 1 edges will be removed.

Exactly 2 new edges will be added.

Insertion of Point p_i



Algorithm

— Compute $e = \text{edge}(p_i)$, and find set Y of all edges that will be destroyed.

— Update circular link list of edges of **CH**.

— For each $p \in \left(\bigcup_{e \in Y} X(e)\right)$:

Update flag(p) and edge(p).

—If e' and e'' are two new edges created, then compute $X(e')$ and $X(e'')$.

Running Time of i^{th} iteration

- Number of edges of convex hull that are destroyed. — $O(|Y_i|)$
- Number of new edges of convex hull that are created. — $O(1)$
- Number of points in the cones that get destroyed. — $O(|W_i|)$

On summing these over all i (see below), we get expected time complexity is $O(n \log n)$.

Let

Y_i = Set of edges which are deleted on addition of point p_i

W_i = Set of those points p for which edge(p) changes on add^n of p_i

Fact 1: $\sum_{i=4}^{n} |Y_i| \leq 2n$, because total no. of edges added is $\leq 2n$.

Fact 2:

$$\text{Exp}\left(\sum_{i=4}^{n} W_i\right) = \sum_{p \in S} \text{Expected no. of times edge}(p) \text{ changes} = O(n \log n)$$