清华算法营-5A

06/10/2018

邓俊辉

deng@tsinghua.edu.cn

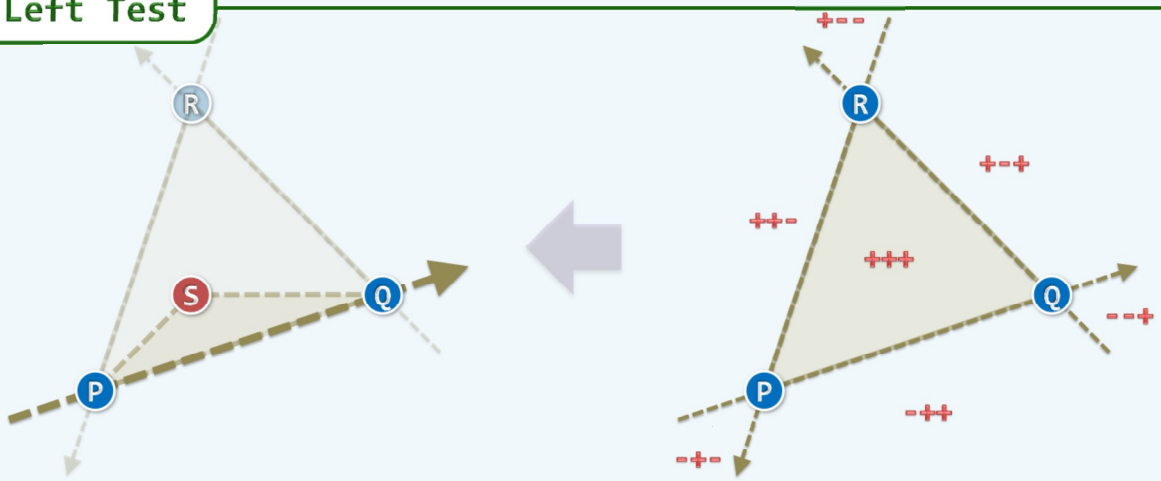
In-Triangle Test


```
bool InTriangle( P, Q, R, S )

   return ToLeft( P, Q, S ) == ToLeft( Q, R, S )

       && ToLeft( Q, R, S ) == ToLeft( R, P, S );
```

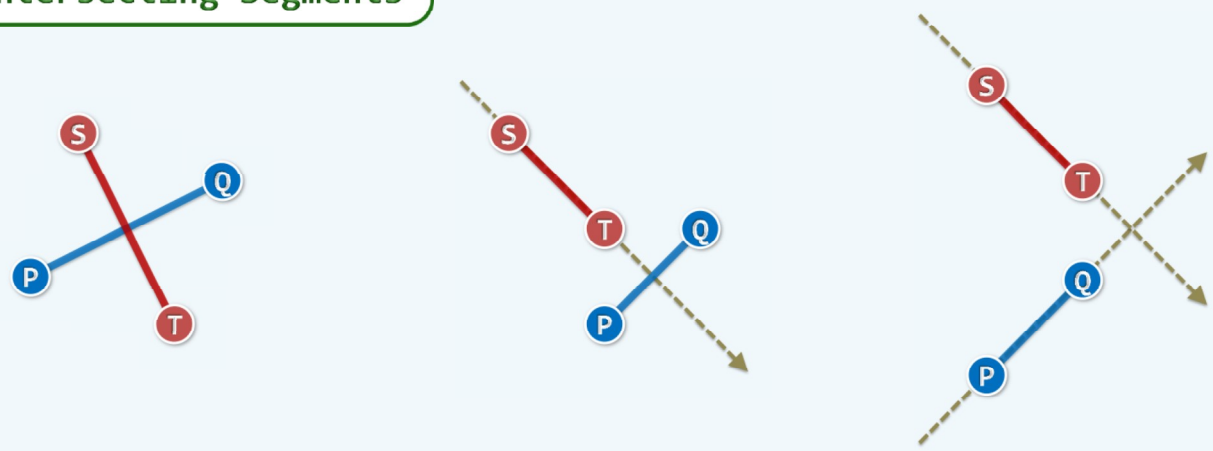
To-Left Test




```
bool ToLeft( P, Q, S )
   return  0 < P.x * Q.y - P.y * Q.x
             + Q.x * S.y - Q.y * S.x
             + S.x * P.y - S.y * P.x;
```

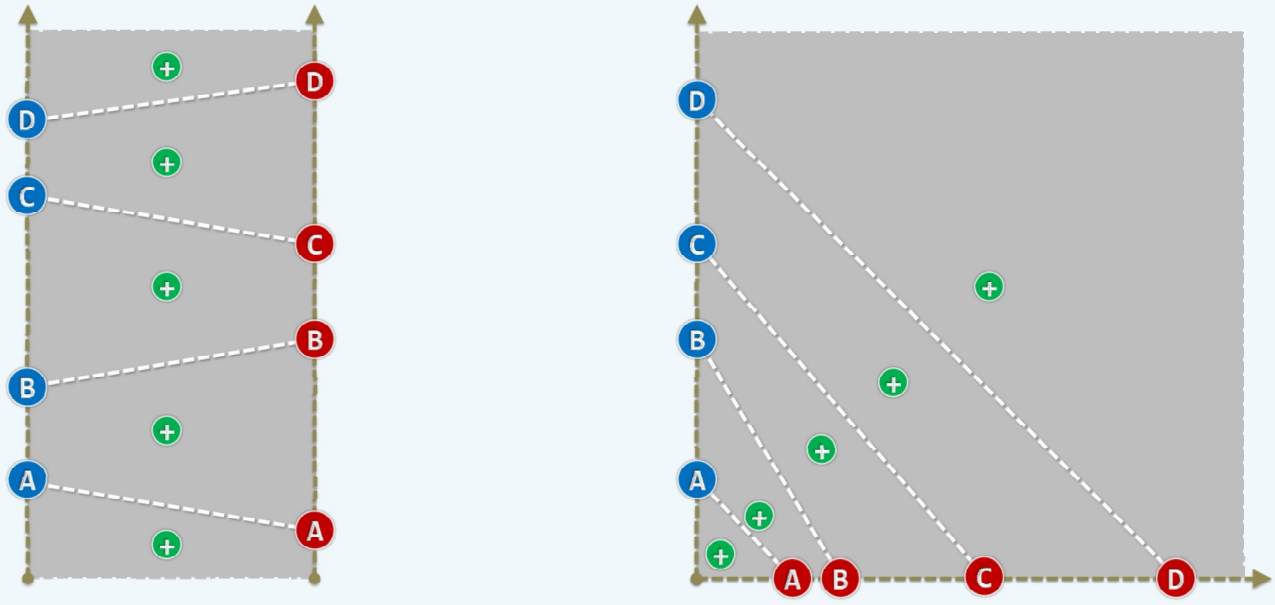
$$2 \cdot S_{\Delta PQS} = \begin{vmatrix} P_x & P_y & 1 \\ Q_x & Q_y & 1 \\ S_x & S_y & 1 \end{vmatrix}$$

Intersecting Segments

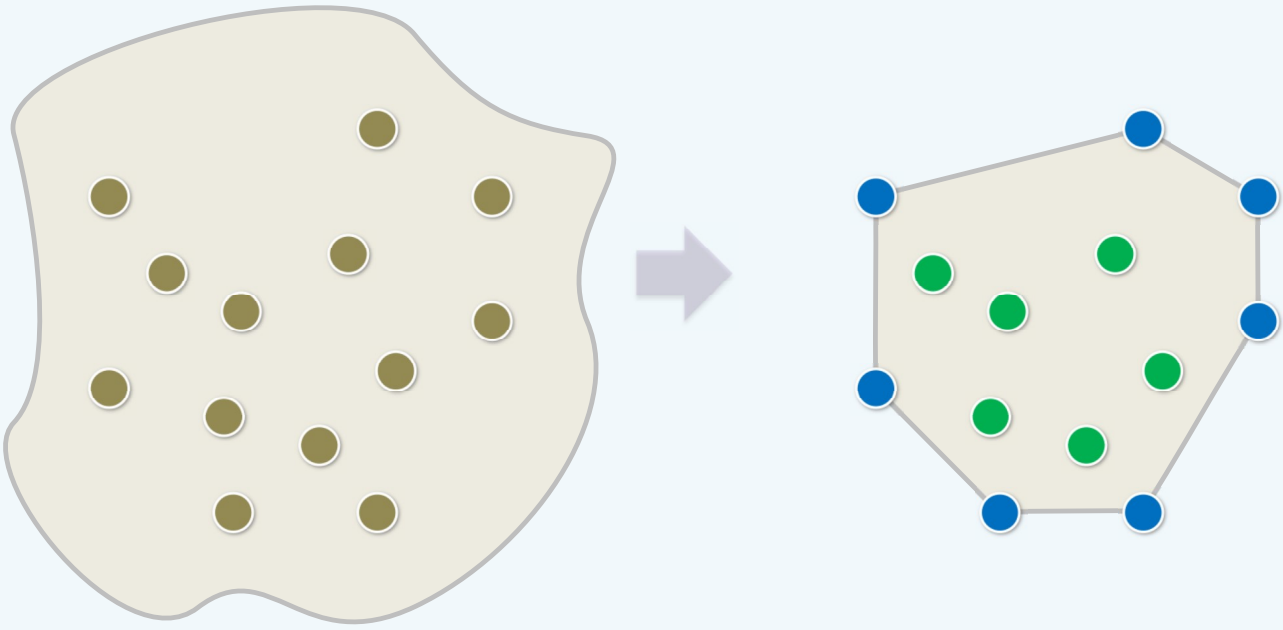



```
bool intersect( P, Q, S, T )
   return ToLeft( S, T, P) ^ ToLeft( S, T, Q)
       && ToLeft( P, Q, S) ^ ToLeft( P, Q, T);
```

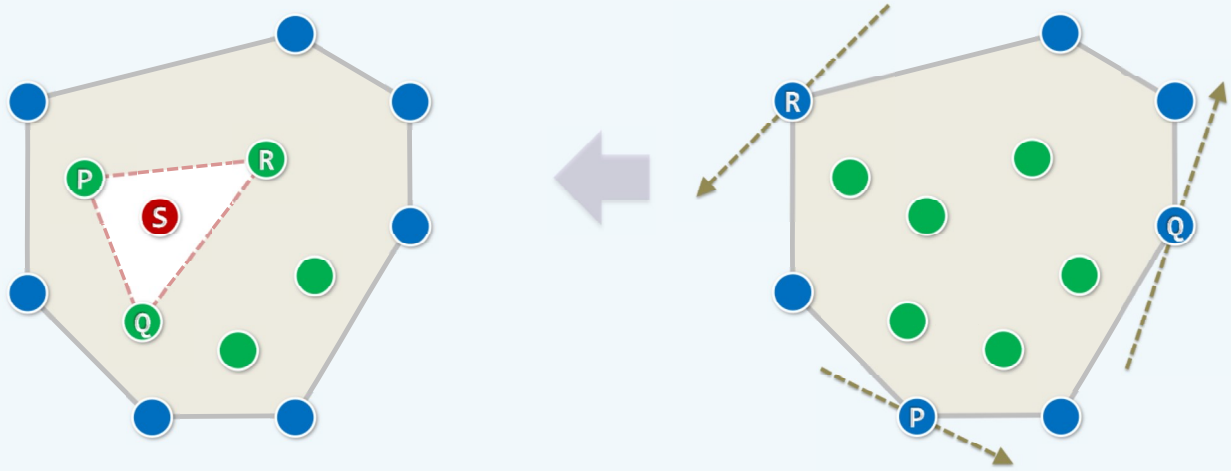
Point Location in A Ladder



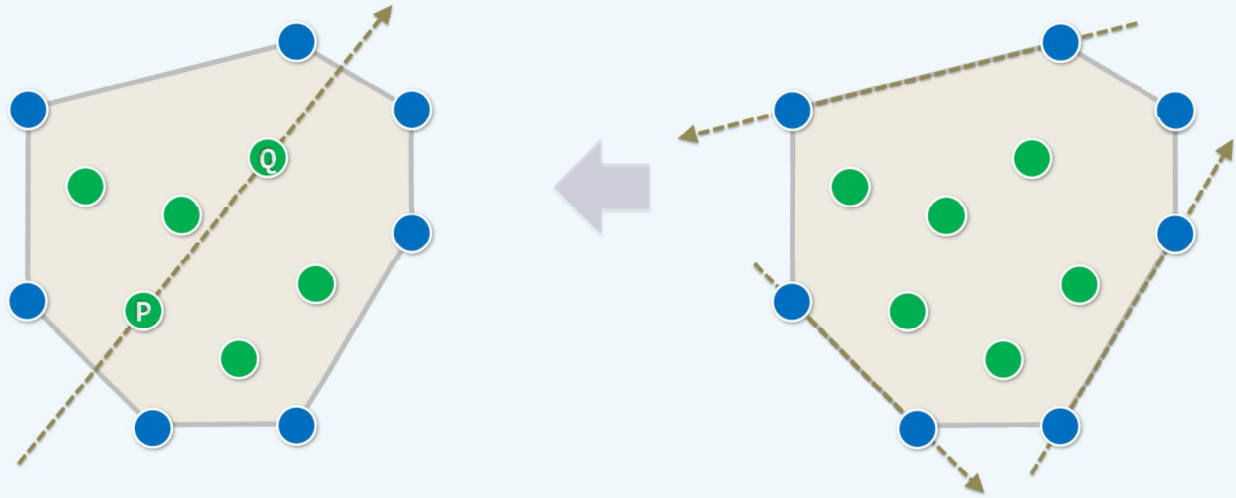
CH: Convex Hull



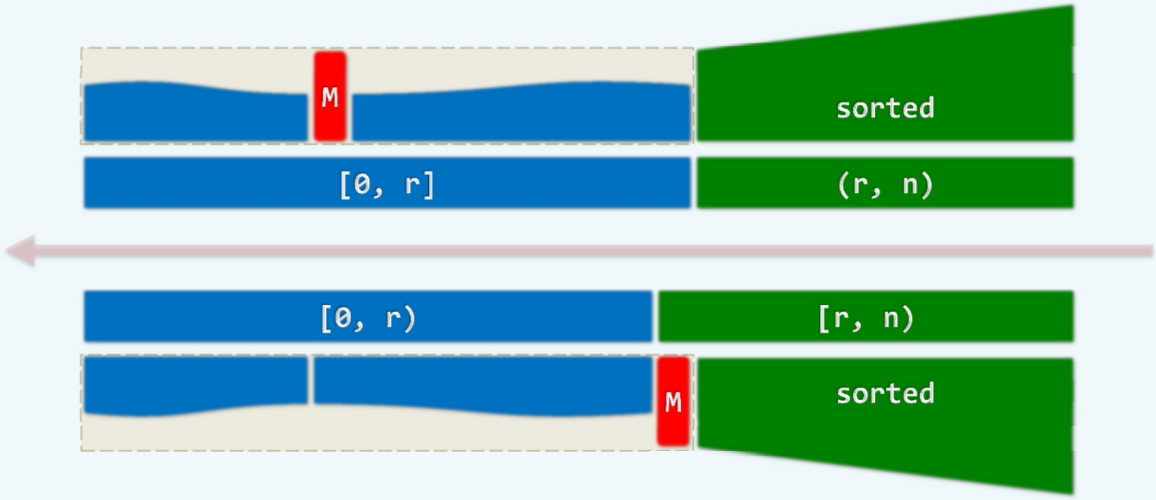
CH: Extreme Points



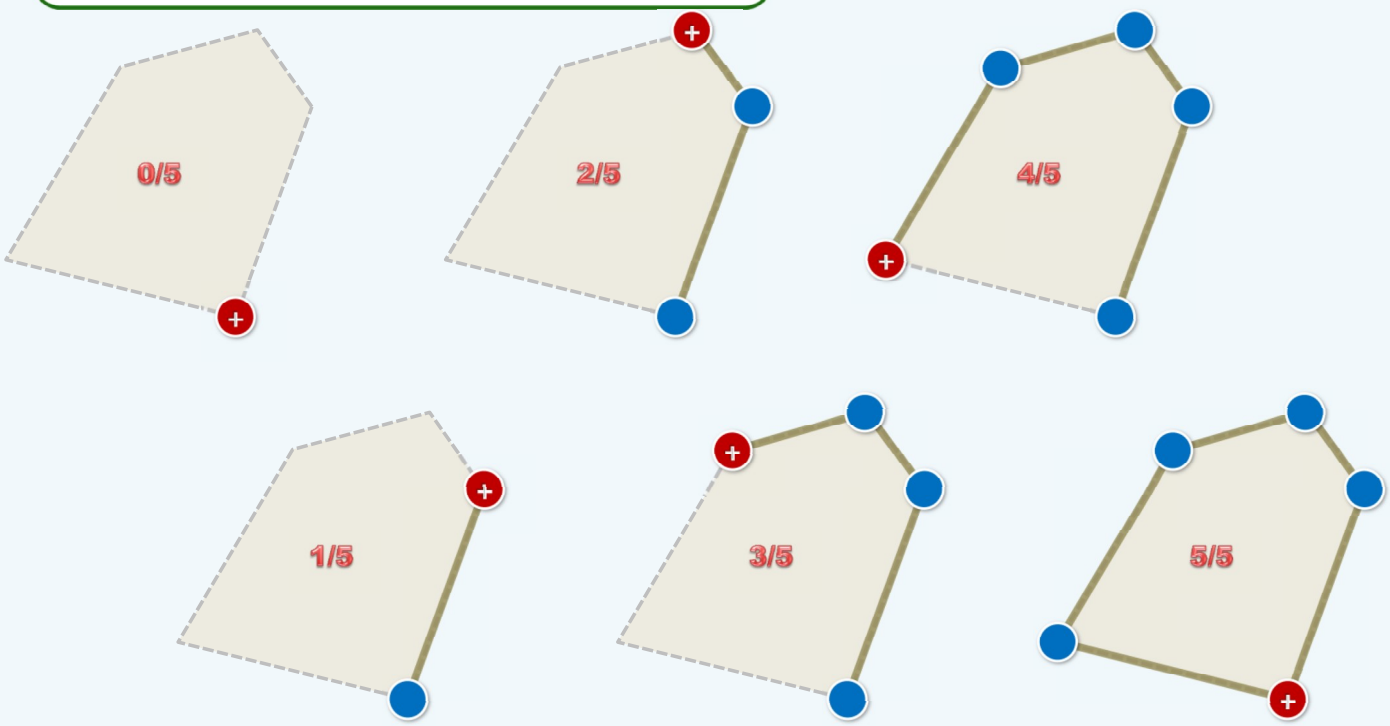
CH: Extreme Edges



CH: Jarvis March: Selectionsort

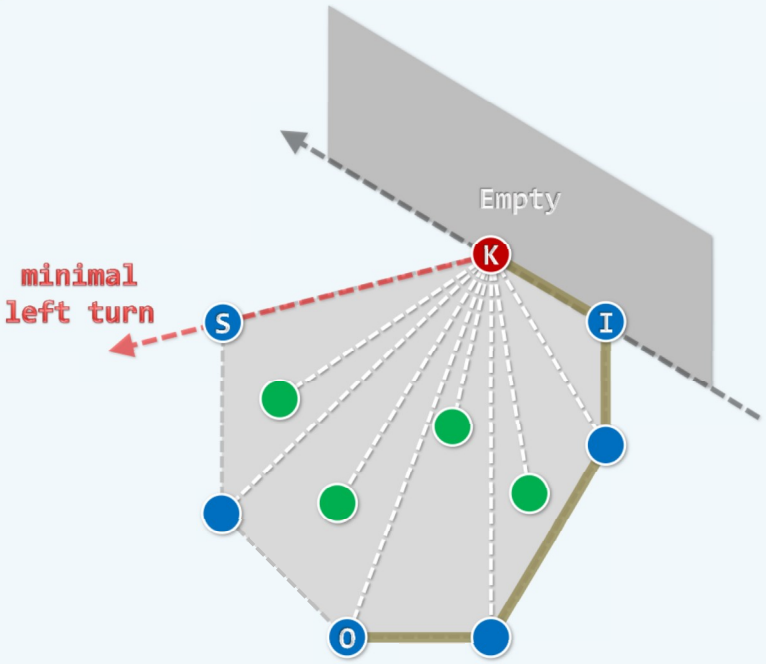


CH: Jarvis March = Gift Wrapping

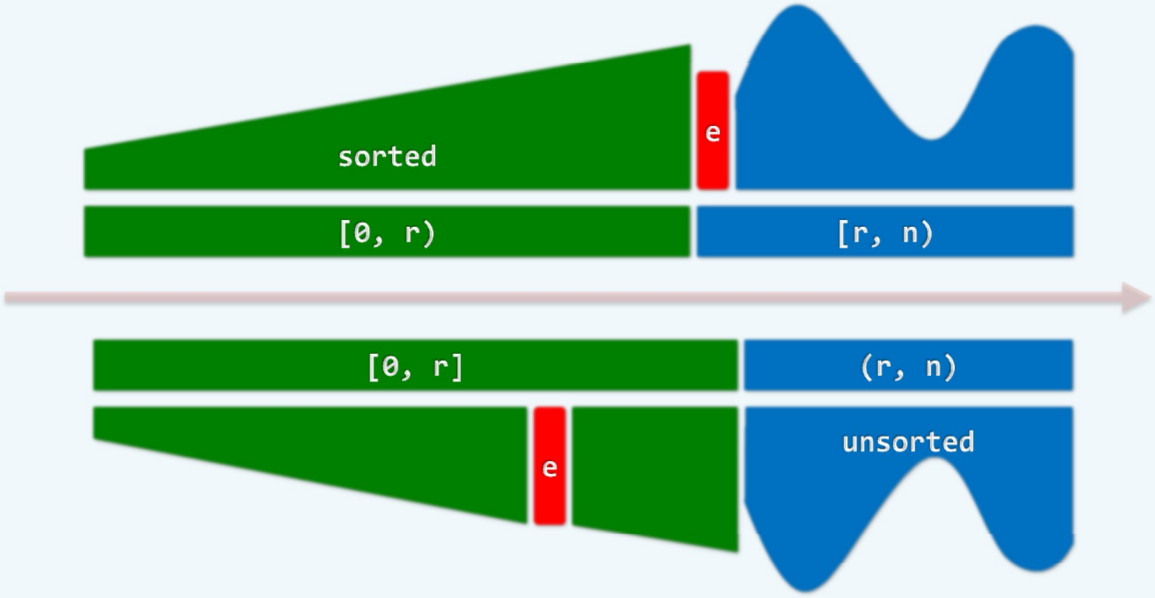


CH: Jarvis March: Algorithm

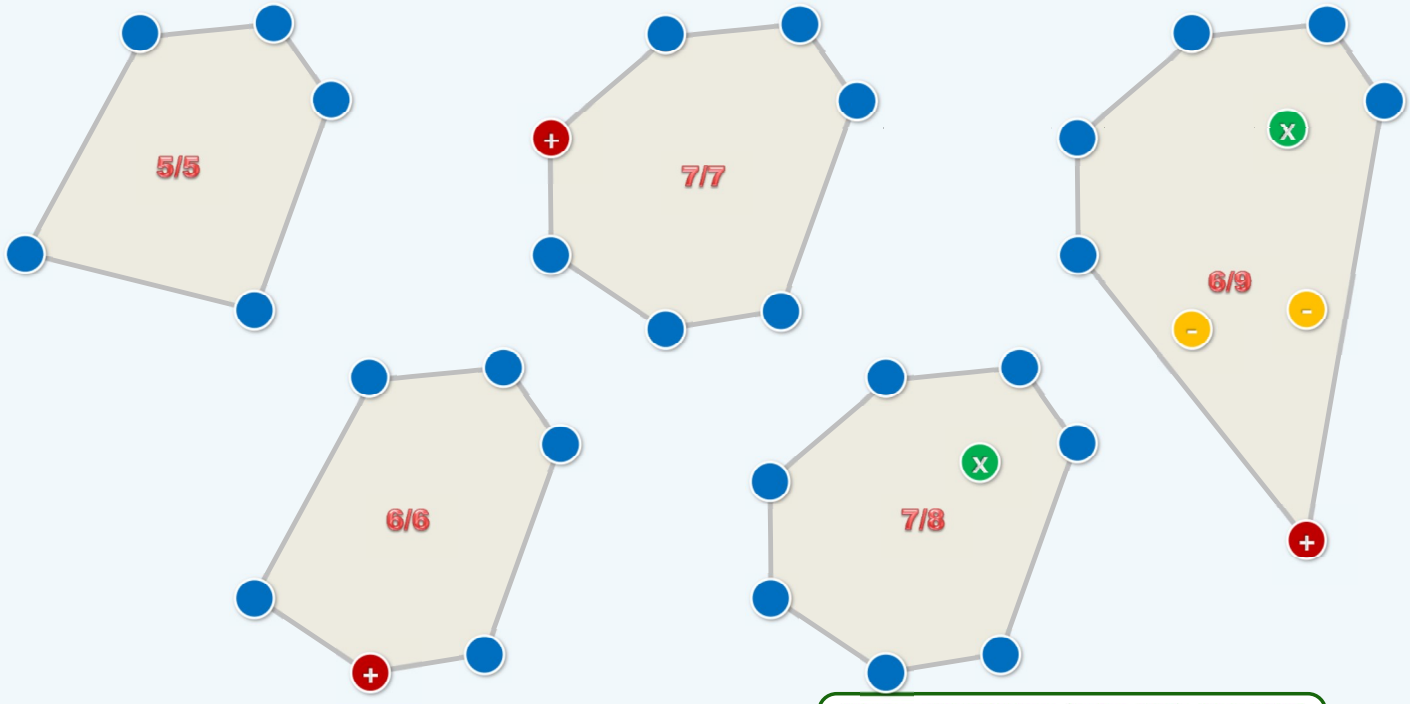
```
Find O an extreme point
Find OK the first extreme edge
Repeat
    Find the next extreme edge KS
    K = S
until K == O
//What if S is NOT unique?
```



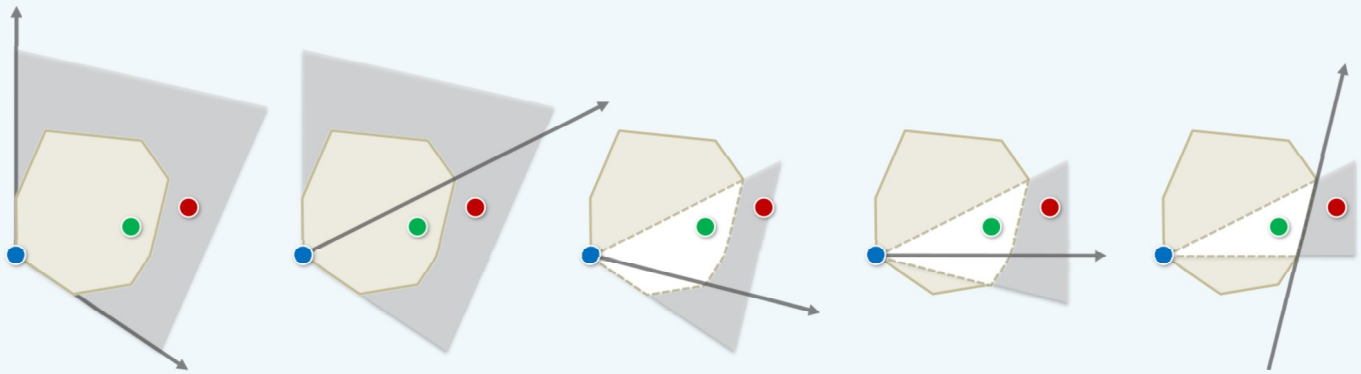
CH: Incremental Construction: Insertionsort



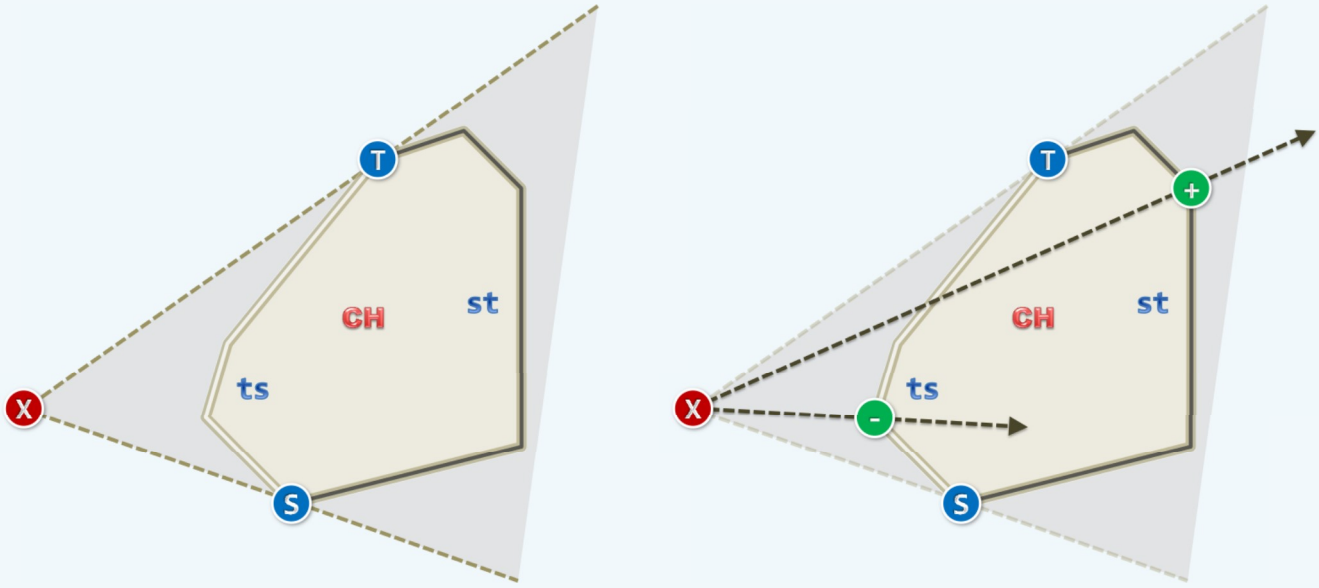
CH: Incremental Construction: Strategy



CH: Incremental Construction: In-Convex-Polygon Test



CH: Incremental Construction: Support Lines + Turn Pattern



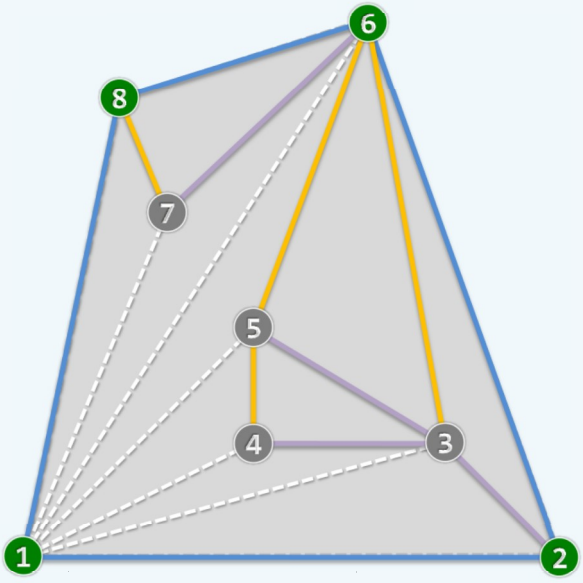
CH: Graham Scan: Complexity

Planar graph

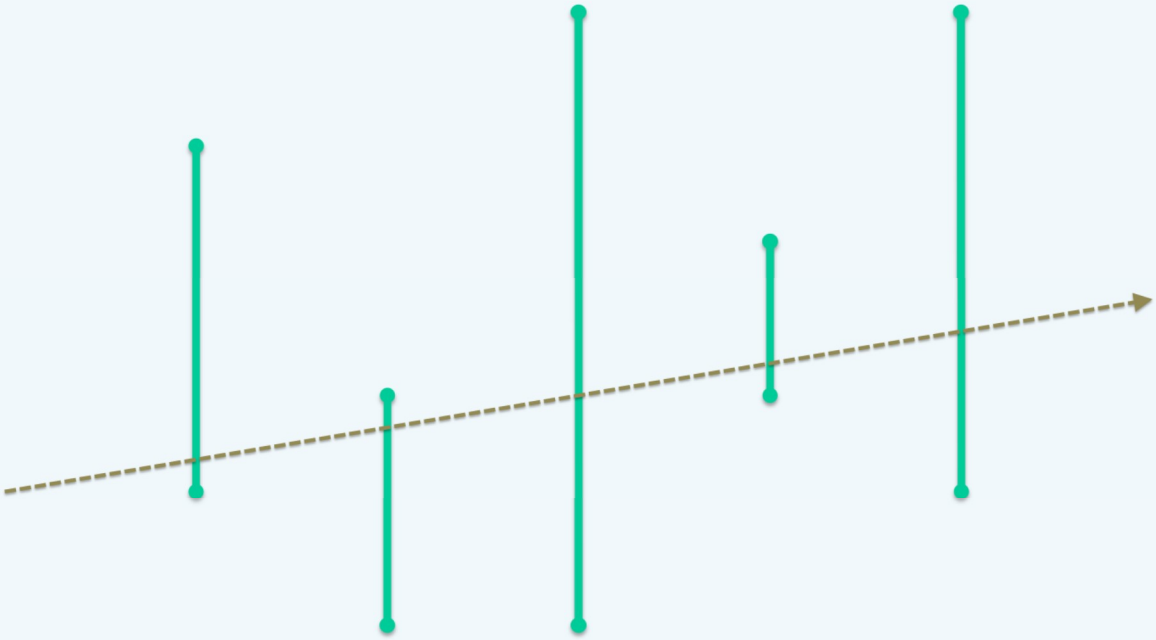
$$v - e + f - c = 1$$

$$e = \mathcal{O}(n)$$

$$f = \mathcal{O}(n)$$



CH: Fruit Ninja



CH: Fruit Ninja: Intersecting Convex Hulls