OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

Dynamic convex hull

Radek Matějka (matejra4)

CTU in Prague, FEL - A4M39VG

2012

Content

Problem definition

Dynamic convex hull problem

Maintain a convex hull allowing the following operations:

- point insertion
- point deletion

- Presented solution is from **Mark H. Overmars and Jan Van Leeuwen** [3].

Problem decomposition

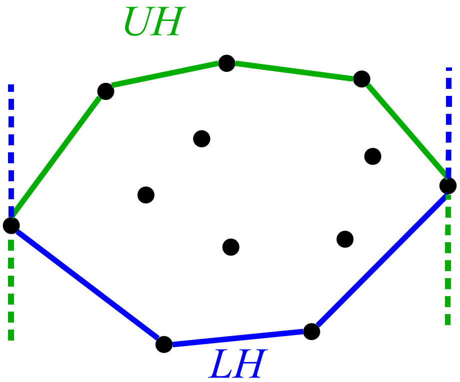


IMAGE FROM [2]

- Splitting to upper hull and lower hull.
- Only upper hull(UH) will be considered.

Data structure overview

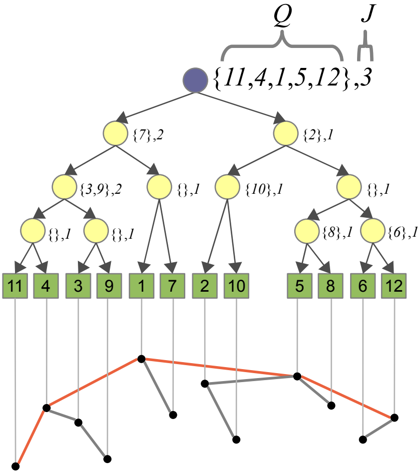


IMAGE FROM [2]

- balanced tree
- root represents whole upper hull
- inner node represents upper hull of its subtree
- inner node contains
  - Q - concatenable queue
  - J - splitting index
- leaf represents point

Concatenable queue

Concatenable queue definition

Operations and required complexity:

- search
- split
- splice
- insert

each in O(log(n))

Concatenable queue is used to store indices to points in inner nodes. The points are **sorted** according to x coordinate. The stored points are points (necessarily not all) of some particular hull (eg. portion of a hull of a subtree).

Insertion/deletion overview

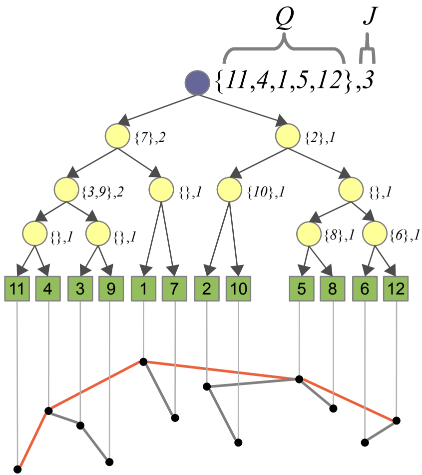


IMAGE FROM [2]

insertion/deletion:

1. descend
2. insert/delete leaf
3. ascend

Descend

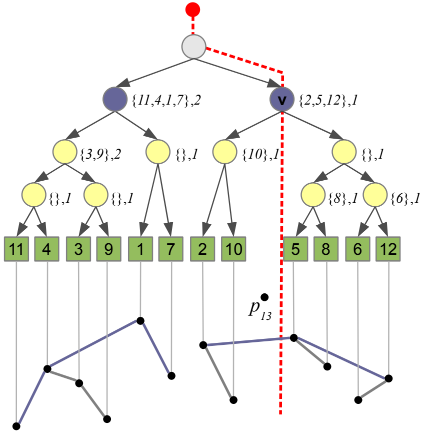
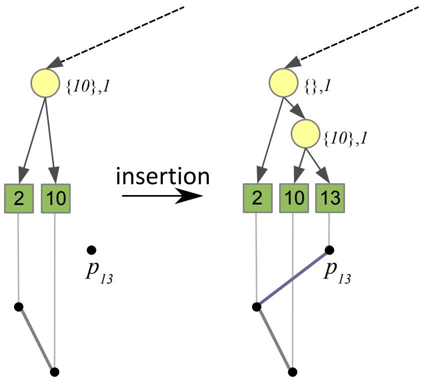


IMAGE FROM [2]

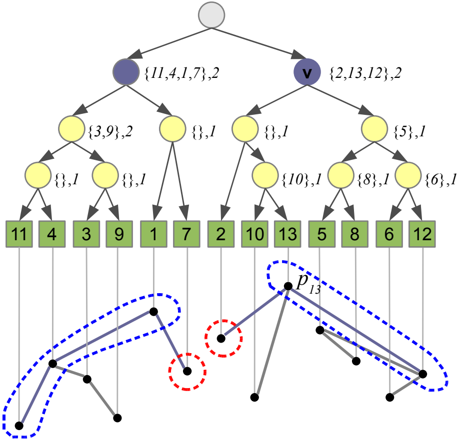
function DESCEND(TreeNode v, Point p)
    if $v \neq LEAF$ **then**
        $Q_L, Q_R \leftarrow split(v.Q, v.J)$
        $v.lChld.Q \leftarrow splice(Q_L, v.lChld.Q)$
        $v.rChld.Q \leftarrow splice(v.rChld.Q, Q_R)$
        if $p.x \leq v.rChld.minX$ **then**
            $v \leftarrow v.lChld$
        else
            $v \leftarrow v.rChld$
        end if
        descend(v, p)
    end if
end function

Insert point



- Once the position is found you can perform required operation.
- In our case, we insert a new point. Deletion would be done analogicaly.

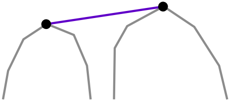
Ascend



Q_1:={11,4,1} Q_2:={7} Q_3:={2} Q_4:={13,12} J:=3

IMAGE FROM [2]

- Goal: ascend the tree and splice two nonoverlapping upper hulls of siblings.



- Special function needed - **bridge(..)**

Bridge (1)

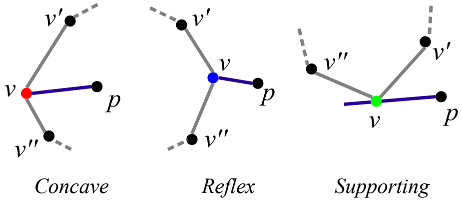


IMAGE FROM [2]

Concave v is concave if vp crosses inside of a hull, otherwise it is reflex or supporting

Reflex v is reflex if v' and v" are on different sides of a line induced by vp

Supporting v is supporting if v' and v" are on the same side of a line induced by vp

Bridge (2)

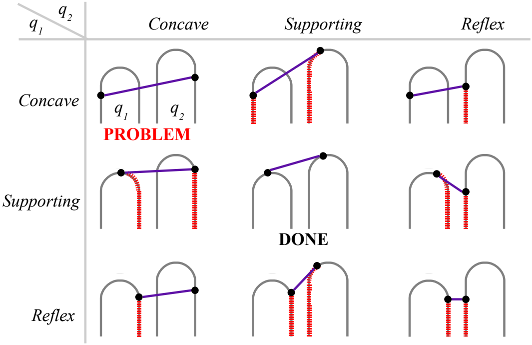


IMAGE FROM [2]

- In all cases except one (noted as problem) we can immediately prune the search space.
- The concave-concave case requires special approach.

Bridge (3)

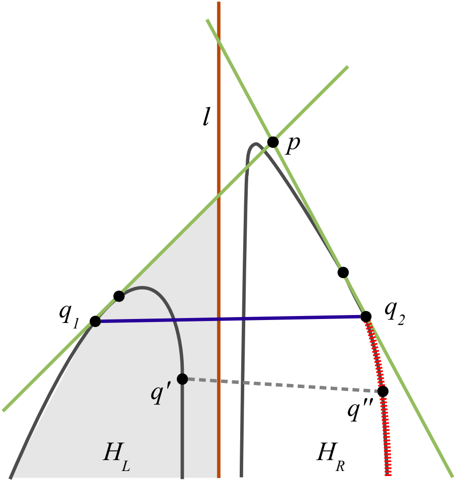


IMAGE FROM [2]

- In concave-concave case a special investigation is needed.
- Construct tangents of hulls at points q_1, q_2 respectively.
- Examine the intersection (point p).
- If p is on the right throw away points under q_2. (figure)
- If p is on the left throw away points under p_1.

Ascend - summary

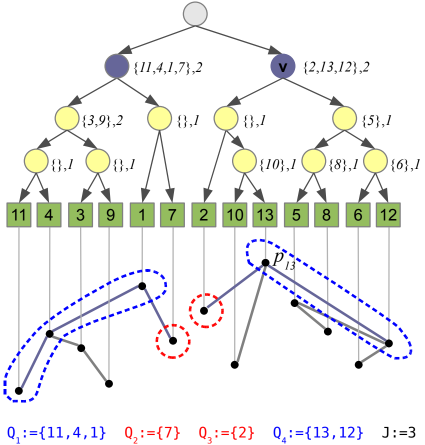


IMAGE FROM [2]

function ASCEND(TreeNode v)
  if $v \neq ROOT$ **then**

$Q_1, Q_2, Q_3, Q_4, J \leftarrow bridge(v.Q, v.sibling.Q)$
  $v.father.Q \leftarrow splice(Q_1, Q_4)$
  $v.father.J \leftarrow J$
  $v.father.lSon.Q \leftarrow Q_2$
  $v.father.rSon.Q \leftarrow Q_3$
  $ascend(v.father)$
  end if
end function

Complexity

Complexity of Overmars and van Leeuwen's algorithm

Inserts and deletes in $O(log^2(n))$

$$\Big(\overbrace{log(n)}^{ascend} \cdot \overbrace{log(k)}^{split} + \overbrace{log(n)}^{descend} \cdot \overbrace{log(m)}^{bridge}\Big) \in O(log^2(n))$$

$$k \leq n; m \leq n$$

n - total number of points
k - number of points in queue being spliced
m - number of points in both hulls when bridging

Conclusion

- If deletion or insertion is not needed there are faster algorithms.
- Faster choice can be Brodal and Jacob working in $O(n \cdot log(n))$.

References

- [1] *Franco P. Preparata, Michael Ian Shamos* - Computational Geometry: An Introduction
- [2] *Jan Novák* - Maintaining 2D Convex Hulls
- [3] *Mark H. Overmars, Jan Van Leeuwen* - Maintenance of Configurations in the Plane

OI-OPPA. European Social Fund
Prague & EU: We invest in your future.