AuW-pg06-bf

In dieser Aufgabe entwickeln wir einen Algorithmus, der die konvexe Hülle einer Menge von n Punkten in der Ebene mit der Divide-and-Conquer-Technik berechnet. In der gesamten Aufgabe dürfen Sie annehmen, dass keine drei Punkte auf einer Geraden liegen, und dass keine zwei Punkte die selbe x-Koordinate haben. Weiter dürfen Sie ohne Beweis verwenden, dass der Median einer Liste von k Zahlen in Zeit $O(k)$ gefunden werden kann.

(a)

Seien P_1 und P_2 zwei konvexe Polygone mit insgesamt n Punkten, gegeben im Gegenuhrzeigersinn, welche durch eine vertikale Gerade voneinander getrennt sind. Zeigen Sie, dass die konvexe Hülle von $P_1 \cup P_2$ in Zeit $O(n)$ berechnet werden kann.

We are given two convex polygons P_1 and P_2 with a total of n points, given in counterclockwise order, and separated by a vertical line. We need to show that the convex hull of $P_1 \cup P_2$ can be computed in $O(n)$ time.

Algorithm to Merge Two Convex Hulls:

1. **Find Upper and Lower Tangents**:
   - **Upper Tangent**: Start with the rightmost point of P_1 and the leftmost point of P_2. Adjust the points to find the upper tangent.
   - **Lower Tangent**: Similarly, find the lower tangent starting with the rightmost point of P_1 and the leftmost point of P_2.
2. **Merge the Hulls**:
   - Remove points that are inside the tangents.
   - Concatenate the points of P_1 and P_2 from the tangents.

Steps in Detail:

1. **Initialize**:
   - Let $p =$ rightmost point of P_1.
   - Let $q =$ leftmost point of P_2.
2. **Find Upper Tangent**:
   - While there exist points above the line joining p and q in P_1, move p counterclockwise.
   - While there exist points above the line joining p and q in P_2, move q clockwise.
   - Repeat the adjustments until no more moves can be made.
3. **Find Lower Tangent**:
   - While there exist points below the line joining p and q in P_1, move p clockwise.
   - While there exist points below the line joining p and q in P_2, move q counterclockwise.
   - Repeat the adjustments until no more moves can be made.
4. **Merge**:
   - Include points from P_1 from the upper tangent to the lower tangent.
   - Include points from P_2 from the lower tangent to the upper tangent.

Complexity Analysis:

- Each point is visited a constant number of times during the tangent finding process.

- Hence, the merging process runs in $O(n)$ time.

(b)

Verwenden Sie Ihren Algorithmus aus (a) um einen Divide-and-Conquer-Algorithmus zu konstruieren, der die konvexe Hülle einer Menge von n Punkten in der Ebene in Zeit $O(n \log n)$ berechnet.

Use the result from part (a) to construct a divide-and-conquer algorithm for finding the convex hull of n points in $O(n \log n)$ time.

Algorithm:

1. **Base Case**: If there are 1 or 2 points, the convex hull is the points themselves.
2. **Divide**: Divide the points into two equal halves by the median x-coordinate.
3. **Conquer**:
   - Recursively compute the convex hull for the left half.
   - Recursively compute the convex hull for the right half.
4. **Combine**:
   - Merge the two convex hulls using the algorithm from part (a).

Steps in Detail:

1. **Sort Points by x-coordinate** (if not already sorted):
   - Use a linear time selection algorithm to find the median, which ensures $O(n)$ time.
   - Partition the points into two halves based on the median x-coordinate.
2. **Recursive Computation**:
   - Compute the convex hull for the left half.
   - Compute the convex hull for the right half.
3. **Merge**:
   - Combine the two convex hulls using the merge algorithm from part (a).

Complexity Analysis:

- **Divide Step**: Finding the median and partitioning takes $O(n)$ time.
- **Conquer Step**: Solving two subproblems of size $n/2$.
- **Combine Step**: Merging two convex hulls takes $O(n)$ time.

Using the Master Theorem for divide-and-conquer recurrences:

- $T(n) = 2T(n/2) + O(n)$.

This recurrence solves to $T(n) = O(n \log n)$.

Proof of Correctness:

- Each step of the algorithm maintains the properties of the convex hull.
- Base cases are trivially correct.
- The divide step correctly partitions the problem.
- The merge step is proven to be correct in part (a).

Thus, the divide-and-conquer algorithm correctly computes the convex hull in$O(n \log n)$time.