Geometric Algorithms
Homework Exam 1

Alexander Apers
6272932

Utrecht University
a.p.apers@uu.nl

November 23, 2022

Question 1

Definitions

Let A denote the set $\{r\} \cup B$
Let $\mathcal{CH}(A)$ denote the convex hull of point set A (including interior and edges)
Let CH(A) denote the set of vertices that make up the edges of $\mathcal{CH}(A)$

Geometric Observations

Since there is only one red point, it is clear that all bichromatic triangles contain exactly one red point and two blue points. We start with the following observation.

Theorem 1. *The maximum area bichromatic triangle must necessarily consist of 2 blue vertices, b_1, b_2 such that $b_1, b_2 \in$ CH(A).*

Proof. Suppose for contradiction that $b_1 \notin$ CH(A). We denote the maximum area triangle $\triangle rb_1b_2$. We will show that the area of $\triangle rb_1b_2$ can be increased by using a vertex $x \in$ CH(A) instead of b_1. Imagine the line l parallel to the line going through r and b_2 which goes through the point b_1. If we replace b_1 with a point on the other side of l compared to where r and b_2 are, the area of $\triangle rb_1b_2$ will increase. l must cross the boundary of the convex hull in 2 locations since $b_1 \in \mathcal{CH}(A)$ but $b_1 \notin$ CH(A). Therefore, it's guaranteed that b_1 can be replaced by at least one point $x \in$ CH(A) so that the area of $\triangle rxb_2 > \triangle rb_1b_2$. Hence, $\triangle rb_1b_2$ is not a maxmium area triangle. This is a contradiction. We conclude that $b_1 \in$ CH(A) to be a maximum area triangle. □

Similarly, it can be proven that $b_2 \in$ CH(A).

Theorem 2. *Given point r and a point $b_1 \in$ CH(A), and line l going through them, TriangleArea($\triangle rb_1b_2$) is a unimodal function for $b_2 \in$ CH(A) on one side of l.*

Proof. The area of $\triangle rb_1b_2$ □

Algorithm Description

Finding the vertices that make up the boundary of the convex hull can be done in $O(n \log n)$ time complexity. These vertices will be given in clockwise (or counterclockwise) order.

Algorithm 1: Maximum Area Bichromatic Triangle

Input : r, B, CH(A) as an array sorted clockwise of size m
Output: Maximum Area Bichromatic Triangle
MaxArea $\leftarrow 0$
MaxAreaTriangle $\leftarrow \triangle$
for $k \leftarrow 1$ **to** m **do**
    $b_1 \leftarrow$ CH(A)[k]
    $l \leftarrow$ line through r and b_1
    `// find 2 consecutive vertices` $i, j \in$ `CH(`A`)` `such that` i `and` j `are`
    `// on opposite sides of` l.
    $i, j \leftarrow$ BinarySearch(l, CH(A))
    $b_{2\text{left}} \leftarrow$ TernarySearch(CH(A)[$1, ..., i$], AreaTriangle)
    $b_{2\text{right}} \leftarrow$ TernarySearch(CH(A)[$j, ..., m$], AreaTriangle)
    $b_{\text{max}} \leftarrow$ max($b_{2\text{left}}, b_{2\text{right}}$)
    if AreaTriangle($\triangle r b_1 b_{\text{max}}$) $\geq$ MaxArea **then**
        MaxArea $\leftarrow$ AreaTriangle($\triangle r b_1 b_{\text{max}}$)
        MaxAreaTriangle $\leftarrow \triangle r b_1 b_{\text{max}}$
    end
end
return MaxAreaTriangle

Algorithm Correctness

Running Time Analysis

The outer loop loops over every vertex $b \in$ CH(A). This is an $O(n)$ number of vertices in CH(A). Binary search to find the two consecutive vertices that are on opposite sides of l can be done in $O(\log n)$. Applying TernarySearch twice on points on the boundary of the convex hull can be done in $O(\log n)$ time complexity, since evaluating AreaTriangle (calculating the area of a triangle with given vertices) can be done in $O(1)$ time complexity. This means that the whole algorithm will run in $O(n \log n)$ time complexity.