

Problem 1**Part A**

Prove that the intersection of two convex sets is again convex. This implies that the intersection of a finite family of convex sets is convex as well.

Proof. Let A and B be convex sets. Take $p, q \in A \cap B$. Note that both p, q must be in A and B , and since both are convex it follows $\overline{pq} \in A$ and $\overline{pq} \in B$. But this just means $\overline{pq} \in A \cap B$. \diamond

Part B

Prove that the smallest perimeter polygon \mathcal{P} containing a set of points P is convex.

Proof. Assume towards contradiction that \mathcal{P} is the smallest perimeter polygon containing P but is not convex. Then there must exist two points with their line L not contained in \mathcal{P} . We can pick $p, q \in L$ such that p, q are on the boundary of \mathcal{P} . Note that the boundary path connecting p and q cannot be a straight line, otherwise non-convexity would be violated. Therefore the line \overline{pq} provides a strict lower bound on the distance between p and q compared to all other paths, including the current boundary. Therefore by replacing the boundary path between p and q with the line \overline{pq} , the overall perimeter of \mathcal{P} would decrease. However, this means \mathcal{P} could not have been the minimum perimeter polygon, a contradiction. \diamond

Part C

Prove that any convex set containing the set of points P contains the smallest perimeter polygon \mathcal{P} .

Proof. Assume towards contradiction that A is a convex set containing

P that does not contain \mathcal{P} . Consider $A \cap \mathcal{P}$. Since A does not contain \mathcal{P} , there must be a portion of \mathcal{P} outside of A . This portion can be replaced by the corresponding portion of $A \cap \mathcal{P}$. Using (a) and (b) we know that $A \cap \mathcal{P}$ must be a convex polygon. Since it is an intersection of polygons, the perimeter must also be smaller than \mathcal{P} , a contradiction. \diamond

Problem 2

Describe an $O(n \log n)$ time method for determining if two sets, A and B , of n points in the plane can be separated by a line.

The main idea for the following algorithm is that two sets of points are separable by a line if and only if their convex hulls (with their interiors) do not intersect. Since the computed convex hulls are just their boundary, testing for this intersection means checking that

1. None of the edges intersect
2. Neither convex hull is contained within the other

If it is determined that none of the edges intersect, then the convex hulls can only be in 2 configurations: one is contained entirely inside the other, or they are separated. This means that checking if a convex hull is contained in another simplifies to checking if a single point is contained. (★)
In total then, the algorithm is

Algorithm 1 Determine if two point sets A and B are separable

```

1: procedure CANSEPERATE( $A, B$ )
2:    $\mathcal{H}_1 \leftarrow \text{CONVEXHULL}(A)$ 
3:    $\mathcal{H}_2 \leftarrow \text{CONVEXHULL}(B)$ 
4:   if EDGESINTERSECT( $\mathcal{H}_1, \mathcal{H}_2$ ) then
5:     return False
6:   else if  $\exists p \in \mathcal{H}_1$  in  $\mathcal{H}_2$  then
7:     return False
8:   else if  $\exists p \in \mathcal{H}_2$  in  $\mathcal{H}_1$  then
9:     return False
10:  else
11:    return True
12:  end if
13: end procedure

```

The time complexity can be broken down by examining each major line of the algorithm.

- 2 and 3)** Finding the convex hull per set of points takes $O(n \log n)$ using the Graham Scan algorithm.
- 4)** Determining if any of the edges intersect can be done via a sweeping line algorithm in $O(e \log e)$ where e is the total number of edges in both \mathcal{H}_1 and \mathcal{H}_2 . Since the number of edges per convex hull is $O(n)$, EDGESINTERSECT is $O(n \log n)$.
- 6 and 8)** As previously discussed (see (★) above), checking if such a point p exists in either case can be done by just picking any point from one set and then checking if it is contained in the other. This boils down to a winding number problem which can be solved in a single pass over the edges. Therefore the time complexity is just $O(n)$.

The total time complexity then is $O(2n \log n + n \log n + 2n) = O(n \log n)$.