Computational Geometry.

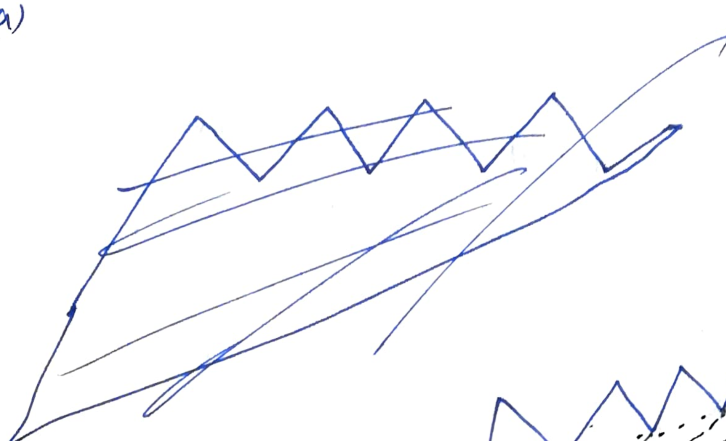
Online-Test 01 Spring 22

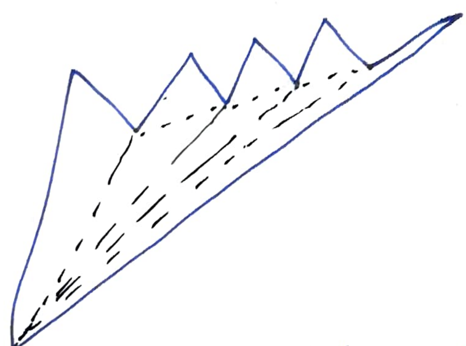
Submitted By:

Pratyush Jaiswal

18EE35014

Q1: (a)





A 10-vertex simple polygon with a unique triangulation.

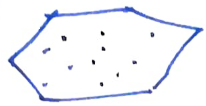
Q1. (b) Given:

The polygon is simple. ⇒ Assuming no holes and ~~&~~ no intersection of edges.

for framing the algorithm, let's assume the vertices of the simple polygon P are given in clockwise direction.
Now, to check the ortho-convex polygon, simply check the turn angles at vertices to be $\pm \frac{\pi}{2}$.
If not then it won't be ortho-convex.
Taking three-consecutive vertices & check the dot-product. It will be done in $\boxed{O(n)}$ Time Complexity & $\boxed{O(1)}$ Space complexity.

Q2.

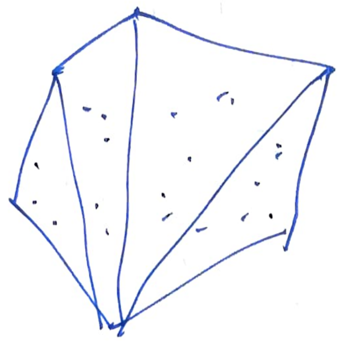
P



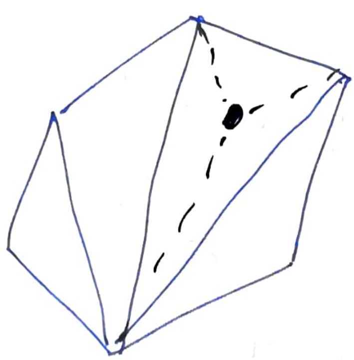
n-points ~~polygon~~

k - convex polygon.

(i). We have k corners in the outer polygon & n-internal points.



The upper figure shows the situation just after $(k-3)$ diagonals are drawn, the lower shows a later situation, in which we want to include the black node.



The black node gives.

- Three extra edges.
- Two extra triangles.

Thus the number of edges will be.

$k+(k-3)+3n = 3n+2k-3.$

Including existing edges.

No. of edges $\Rightarrow$ $3n+k-3$

(ii) - We can find the triangulation of the corners by choosing one of them, & draw an edge to the $k-3$ corners.

- Together with the outer edges this gives $k+k-3=2k-3$ edges and $k-2$ triangles.

- We then take each of inner points, q and do as follows: We find the triangle that q resides in & draw edges to the three corners.

All the triangles will be stored. After checking the required triangle, that triangle will be popped out & new three triangles will be appended.

Like this we proceeds.

Time complexity:

Max Δs at any point $= k+2n-2$

$= O(k+n)$.

Checking for lying inside ~~which~~ triangle $\rightarrow O(1)$

Doing this for all interior points $\rightarrow O(n)$

$\therefore$ Total $\rightarrow O(n \times (k+n))$

$= O(nk+n^2)$

Q3. Using Pick's theorem,

Area(P) = # interior points + (# boundary points)/2 − 1

$$A = I + \frac{B}{2} - 1$$

Area of polygon can be found by drawing horizontal lines to divide into triangles.

For finding B, we can do :-

Consider v_1 & v_2 as two vertices of polygon P

Case I: If edge forming v_1 & v_2 is parallel to X axis

No. of points ⇒ $abs(x_{v_1} - x_{v_2}) - 1$

Case II. If edge parallel to Y-axis

No. of points ⇒ $abs(y_{v_1} - y_{v_2}) - 1$.

Case III: Else we can find by the following.

$GCD(abs(x_{v_1} - x_{v_2}), abs(y_{v_1} - y_{v_2}))$

Hence we can find B.

So, we have got (I+B).

Q5.
(a). The upper & lower chains of P & Q are x-monotone; In $O(m+n)$ time, merge four sorted vertex lists & forms an x-sorted order of m+n vertices. Sweep a vertical line from L→R, thus partitioning the plane into $(m+n-1)$ vertical slabs; the intersection of each slab with P & Q is trapezoid; The trapezoid within slab can be implemented in $O(1)$ time ⇒

$(P \cap Q) \Rightarrow O(m+n)$

After determining if $A(P \cap Q) == A(P) == A(Q)$

→ overlapping.

$A(P \cap Q) = A(P) < P(Q)$

→ contained.

$A(P \cap Q) == 0$

→ disjoint.

(b). ~~Ato~~ Take an intersection point & move ~~leftmost~~ while traversing boundaries CW. & ~~make the~~ consider the edges. This will give the union $P \cup Q$. Since traversing ~~edges~~ nodes ⇒ $O(m+n)$

Q6. (a) Here,

Minimum number of y-monotone polygons
= 3.

Diagonals defined by vertices :-

1 → [24, 15]
2 → [2, 10]

} 2 Diagonals

Justification of minimum number:

Here, the top vertices are (1, 3, 23) → 3
bottom vertices are (11, 16, 9) → 3
merge vertices are (2, 24) → 2
Split vertices are (10, 15) → 2.

Here, looking at above vertices, it is clear that the monotone partition is at least three. because y-monotone polygon must have exactly one top & bottom vertex.

~~So, there will be only~~

And our solution is generating three monotones, so it is indeed minimum.

A 8:

(a) Half Edge → e_1

Origin → v_2

Twin → e_2

Incident Face → f_1

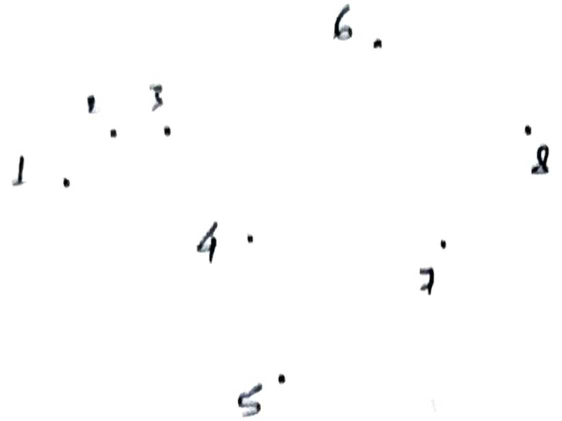
Next → e_3

Prev → e_4

(b). As we know that the line intercepts an edge e_1, so it will also intercept its twin e_1'.

→ From the twin edge, we can get the other face. After following the twin edge & next face, we can get the next edge being intercepted. And this continues until ~~next~~ no events are remaining.

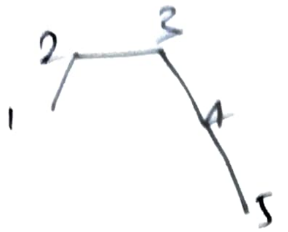
Q 9:



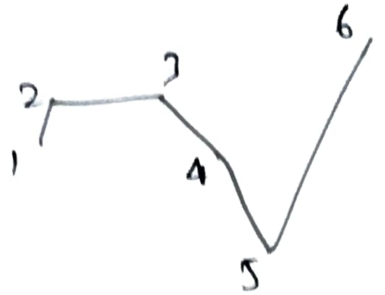
Turn checking needed:

→ Turn check [Right] - 1

→ Turn check [right] → 2

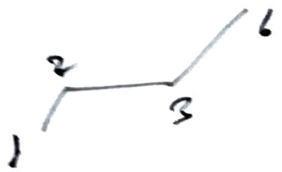


→ Turn check [right] → 5



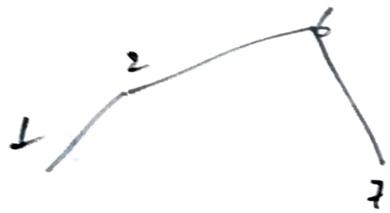
→ Turn check [left] - 4.

→ Turn check. [left] → 5



→ Turn check [left] – 6

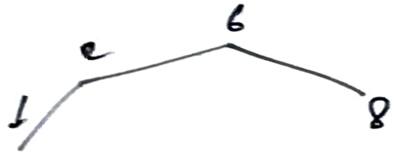
→ Turn check [right) → 7



→ Turn check. [right] → 8

→ Turn check [left] → 9

→ Turn check [right] → 10.



∴ Total checks = 10

& Total left-turn = 4