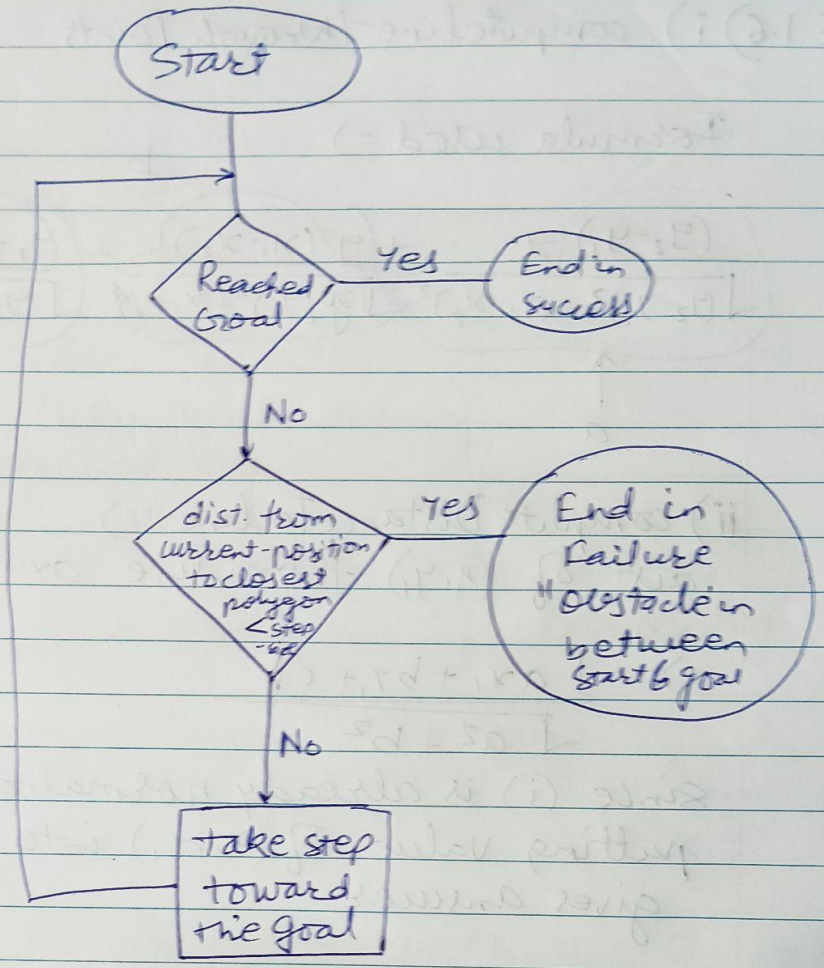
E.1.8) i)



ii) I will replace line 6. I will ask robot to circumnavigate the obstacle and record distance from robot to goal at each instance. Once circumnavigation is done, the robot will be asked to step toward the goal from the point with min dist. from the goal.

functions:) computeTangentToPolygon ⇒ to get tangent once obtacle is hit.

computeDistanceToPolygon ⇒ to get if obstacle is hit or not.

E1.6) i) computeLineThroughPoints

Formula used =)

$$\frac{(y_2-y_1)}{\sqrt{(y_2-y_1)^2+(x_1-x_2)^2}}x + \frac{(x_1-x_2)}{\sqrt{(y_2-y_1)^2+(x_1-x_2)^2}}y + \frac{(y_1x_2-x_1y_2)}{\sqrt{(y_2-y_1)^2+(x_1-x_2)^2}} = 0$$

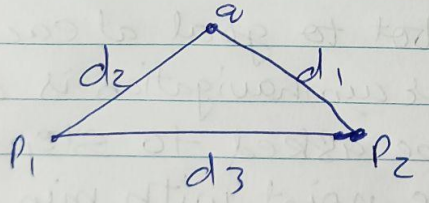
a, b, c

ii) computeDistanceToLine =)
dist of (x_1, y_1) from line $ax+by+c=0$.

$$\Rightarrow \frac{|ax_1+by_1+c|}{\sqrt{a^2+b^2}}$$

Since (i) is already normalized, directly putting value of (x_1, y_1) into eq^n of (i) gives answer.

iii) computeDistancePointToSegment =)
Calculate 3 distance.



if $d_3^2 > d_2^2 + d_1^2$
→ $w = 0$

if $d_2^2 > d_1^2 + d_3^2$
→ $w = 2$

if $d_1^2 > d_2^2 + d_3^2$
→ $w = 1$

Simple application euclidean geometry

E1.7) i) computeDistancePointToPolygon
use computeDistancePointToSegment on each line segment of Polygon to get.

$$\begin{bmatrix} d_1 & d_2 & \cdots & d_n \\ w_1 & w_2 & & w_n \end{bmatrix}$$

for all line segments.
consider distances d_i only if $w_i = 0$,
and then find min of d_is to required value,

if there is not d_i with $w_i = 0$.
then find the min of remaining d_j's.

ii) computeTangentVectorToPolygon
Compute distance of Point from each segment and Point.

Consider segments with $w = 0$. only.
Find min of those. And thus the point on the polygon which is closest.
~~Find perpendicular to the line~~
If point is on segment use its slope for tangent.