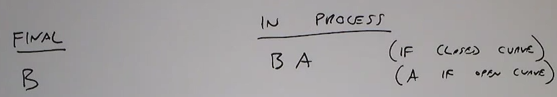
Such kind of pixel level edge linking might be crude so we will look into geometry level edge linking. Geometry level edge linking considers basic polygon to be fit around a set of edged pixels.

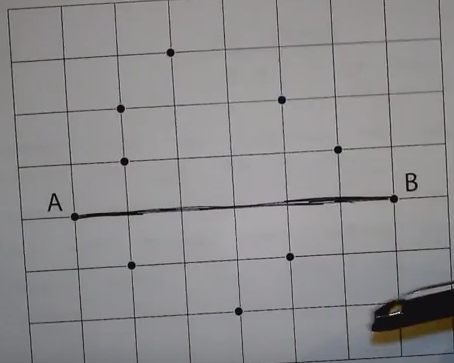
***Polygonal Fit algorithm***:

* Let ‘P’ be a sequence of ordered distinct points (ordered edges) after boundary following.
* Specify two points ‘A’ and ‘B’:
  + If curve is open, ‘A’ and ‘B’ are the natural endpoints.
  + If curve is closed, ‘A’ and ‘B’ are left and rightmost points.
* Specify a threshold ‘T’ which specifies how close the points have to be to one of these boundary lines (pixel distance).

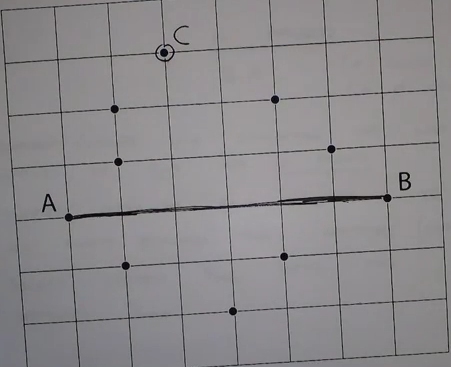
Create two stacks:



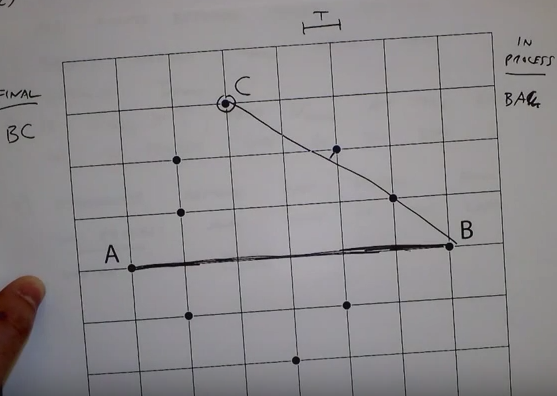
* Compute the line connecting the last vertices of the ‘final’ and ‘in process’ stack.



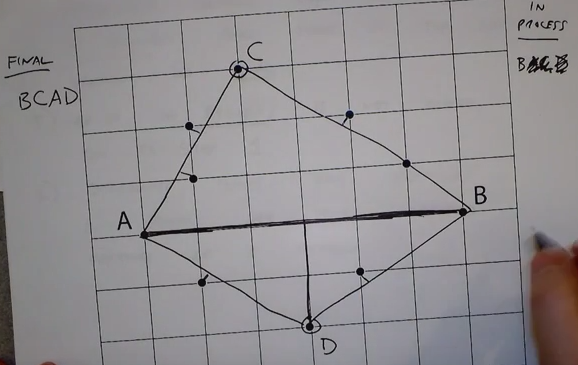
* Compute distances from this line to all points between these vertices (counterclockwise). Select vertex ‘Vmax’ with the maximum distance ‘Dmax’.



* If ‘Dmax’ > T, put ‘Vmax’ at the end of ‘in process’ stack and go to step 1.



* Otherwise remove last vertex from ‘in process’ stack and make it the last vertex of ‘final’ stack.
* If ‘in process’ stack is not empty go to step 1.
* Otherwise if done, the vertices in the ‘final’ stack are the ordered vertices of the polygon.



Fitting Straight Lines: