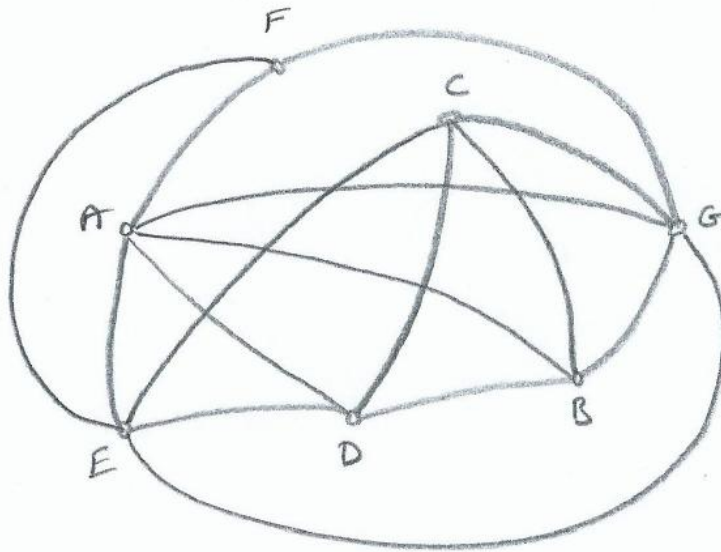


Planarity Algorithm (7 pages; 19/10/26)

A graph is planar if it can be redrawn (if necessary), so that its arcs do not cross.

If a graph contains a Hamiltonian cycle (ie a cycle that passes through all nodes exactly once), then the Planarity Algorithm enables us to establish whether or not the graph is planar.

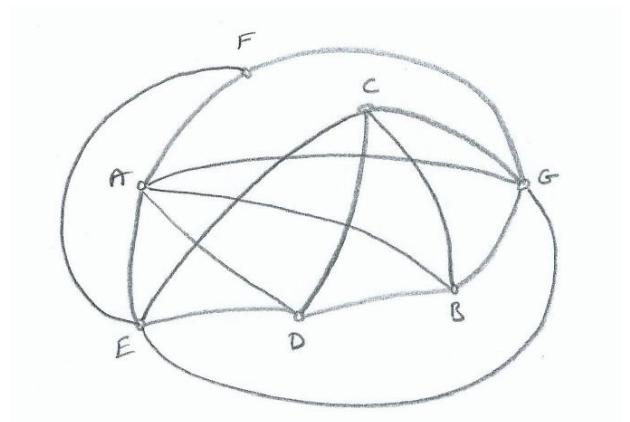


Step 1

Find a Hamiltonian cycle for the above graph.

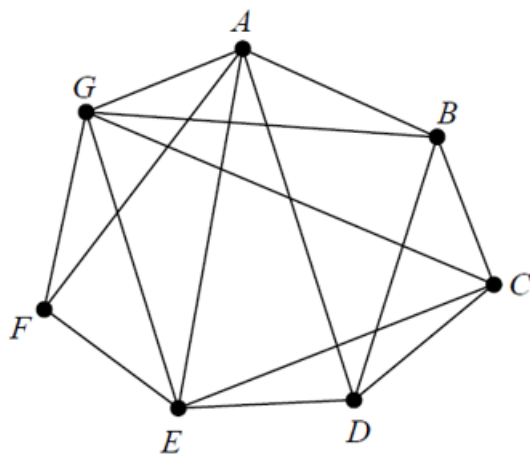
Hamiltonian cycle is

(for example) ABCDEFGA



Step 2

We then draw the polygon ABCDEFGA (ie with the nodes of the Hamiltonian cycle in the correct order), and add in the remaining arcs of the original graph, in such a way that they lie within the polygon:



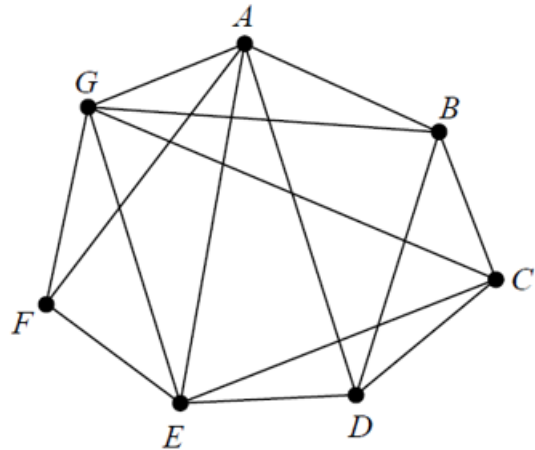
The idea behind the algorithm is that, if the graph is planar, then it will be possible to divide the interior arcs into two groups, such that the members of each group do not cross each other.

Then the arcs of one of the two groups (it doesn't matter which) can be redrawn outside the polygon, and we will have obtained a graph that is isomorphic to the original graph, with no arcs crossing.

For simpler graphs, this grouping could be done by inspection.

AF, AE, AD, BD

GB, GC, GE, EC



But it is likely that an exam question would require the Planarity Algorithm to be followed instead:

Step 3: Create a list of the interior arcs (in any order).

For example: AD, AE, AF, BD, BG, CE, CG, EG

Step 4:

Give the arc AD the label *I*, to indicate that we are allocating it to the group of arcs that will stay inside the polygon.

Step 5

For the remaining arcs in the list, we now consider those that cross AD. [They will have to be allocated to the outside group.]

These arcs are BG, CE & CG.

Step 6

We need to see whether any of these arcs cross each other.

If they do, then we are unable to divide the original arcs into two groups (such that no arcs cross within each group). In that case, the graph is non-planar.

BG, CE & CG do not cross each other, and so they can be labelled *O*.

Step 7

The remaining arcs in the list are AE, AF, BD & EG.

Step 8

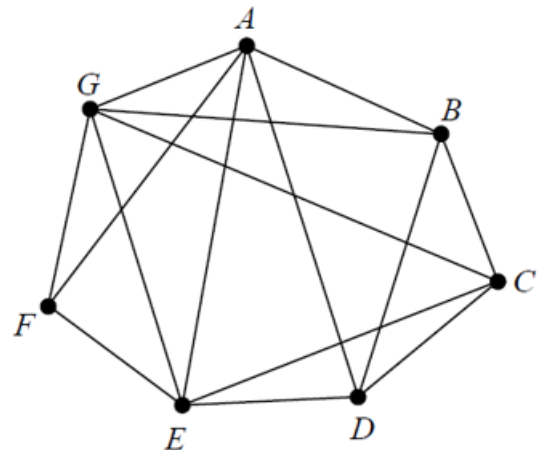
Consider any of these that cross the arcs that have just been labelled *O* (ie BG, CE & CG).

AE: crosses BG (& CG)

AF: crosses BG (& CG)

BD: crosses CE (& CG)

EG: doesn't cross any of them

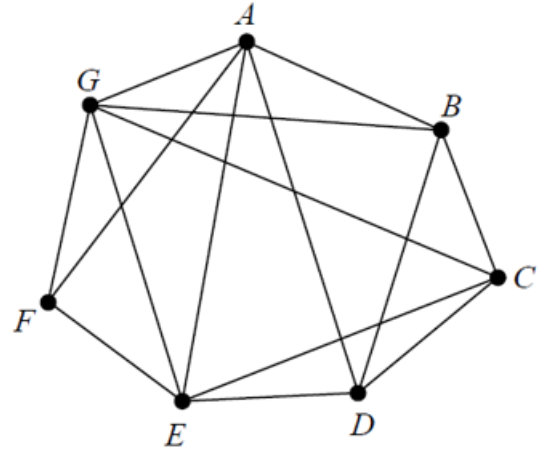


[So arcs AE, AF & BD can't go in *O*.]

Step 9

We need to see whether any of the arcs AE , AF & BD cross each other. [If they do, then they can't go in I either.]

These arcs don't cross each other,
so they can be labelled I .



Step 10

The remaining arc in the list is EG .

Step 11

Does EG cross the arcs that have just been labelled (ie AE , AF & BD)?

Yes – it crosses AF .

Step 12

Label it O (the opposite label to AF).

We have now labelled all the arcs:

AD: I

BG, CE & CG: O

AE, AF & BD: I

EG: O

The best check that the algorithm has been applied correctly is to redraw the graph with arcs placed in their allocated groups, to demonstrate that no arcs cross.

KEY IDEAS

(1) Once some arcs have been given a label (whether I or O), we then consider those arcs that are remaining **that cross the ones just labelled**.

[If there are no arcs that cross the previously labelled arcs, then we can start the process again, and label the first remaining arc I (though it wouldn't matter if we labelled it O).

We then see if these arcs cross each other.

If they do, then the graph is non-planar.

If they don't, then they can be given the opposite label to the arcs labelled previously.

(2) For an arc that has just been given a label, there is never any danger of it crossing another arc that has already been given that label, as all the arcs that crossed any labelled arcs will have been dealt with at an earlier stage.]