SECTION 1 INTRODUCTION

basic terminology
A graph is a set of finitely many points called vertices which may be connected by edges. Figs 1-3 show three assorted graphs.

If there is more than one edge between two vertices (Fig 4) then the graph is said to have multiple edges or parallel edges.

An edge drawn from a vertex back to the same vertex is called a loop. See Fig 5 and also vertex X in Fig 3.

The number of edges connected to a vertex is called the degree of the vertex. In Fig 1, all vertices have degree 3; in Fig 3, Y has degree 1, Z has degree 0 (it's an isolated vertex). A loop is considered to contribute 2 to the degree of a vertex since the edge leaves the vertex and also returns. In Fig 3, X has degree 3.

If the number of edges is E then the sum of the degrees of the vertices is 2E.

This holds because each edge contributes 2 to the total degree count.

A path is a sequence of connected edges (along which you can take a walk through the graph). By convention, a path can repeat a vertex (i.e., a path can cross itself) but can't repeat an edge (no backtracking). In Fig 6, ABCF is a path between A and F; other paths between A and F are AABCF, ABDCF etc. Some paths between A and E are AE, ABE, ABCDBE (which crosses itself at B).

A connected graph is one in which there is a path between any two different vertices (i.e., it's always possible to get from here to there). Figs 1, 4, 5, 6 are connected. Fig 2 is not connected since it's impossible to get from any of A, B, C, D to any of E, F, G, H. Fig 3 is not connected because it's impossible to get to Z from X and Y.

The graph in Fig 2 has two components, i.e., the graph is split into two connected pieces with no edges between the pieces. Fig 3 also has two components.

A path that ends up where it started is called a cycle or a circuit. In Fig 6, ABEA is a cycle; so is the figure eight ABCDBEA. A cycle can repeat a vertex (in fact it must repeat the first and last vertex) but not an edge. A graph with no cycles is acyclic.
the adjacency matrix

A graph can be represented by an adjacency matrix where each entry indicates how many edges there are between vertices. Fig 7 shows a graph and the corresponding adjacency matrix. The entry in row 3, col 4 is 1 because vertices v3 and v4 are connected by one edge. The entries along the diagonal are 0 because no vertex in the graph has a loop. The entry in row 1, col 3 is 2 because there are two edges between v1 and v3.

FIG 7

The adjacency matrix of a graph is symmetric, i.e., the entry in row i, col j is the same as the entry in row j, col i since both entries describe the number of edges between vertices v_i and v_j; equivalently, row i and col i are identical.

isomorphic graphs

Given an adjacency matrix, there are many ways in which the corresponding graph can be drawn. For example, the adjacency matrix

\[
\begin{array}{cccc}
A & B & C & D \\
A & 0 & 1 & 0 & 0 \\
B & 0 & 1 & 1 \\
C & 0 & 1 \\
D & 0 \\
\end{array}
\]

goes with all the graphs in Fig 9. The graphs in Fig 9 are called isomorphic since they have the same adjacency matrix.

FIG 9 six isomorphic graphs

More generally, graphs don't have to be lettered in the same way to be called isomorphic.
In Fig 10, the second graph is isomorphic to the first with

- P corresponding to Q
- Q to B
- R to C
- S to D

And the third graph is also isomorphic to the first, with B playing the role of A and A playing the role of B.

In general: two graphs are isomorphic if you can match up the letters so that the two adjacency matrices are the same.

**PROBLEMS FOR SECTION 2.1**

1. Find the adjacency matrix for this graph

   ![Graph](image)

2. How can an isolated vertex (degree 0) be identified by looking at the adjacency matrix?

3. (a) Let \( S \) be the sum of the degrees of all the vertices in a graph. Show that \( S \) is always even. Suggestion: Look at the connection between \( S \) and the number of edges.

   (b) Use part (a) (among other things) to show that in any graph, there must be an even number of vertices of odd degree. Suggestion: Start like this. Let \( V_1, \ldots, V_k \) be the vertices of even degree.

   Let \( Q_1, \ldots, Q_n \) be the vertices of odd degree. (You want to show that \( n \) is even.) Let \( S \) be the sum of the degrees as in part (a).

   (c) Use part (a) to show that in a group of 25 quarrelsome people it isn't possible for each person to get along with exactly 5 others (assuming that getting along is a 2-way street so that if A gets along with B then B gets along with A).
4. (a good counting problem) To warm up, look at graphs with vertices A, B, C and with no loops or multiple edges.

There are 8 of them:

(Yes the graph with 3 vertices and no edges counts.)
(Yes the graph with 3 vertices and no edges counts.)
(The second, third and fourth graphs are isomorphic but for this problem I'll count them as different.)

Now consider graphs with ten vertices $A_1, \ldots, A_{10}$.

(a) How many are there with no loops and no multiple edges.
(b) How many are there with no loops but now allowing as many as 2 edges between each pair of vertices
(c) How many are there with no multiple edges but now allowing loops, no more than one loop per vertex.

5. For each pair of graphs, decide if they are isomorphic or not.

(a)

(b)

(c)

6. Are there any pairs of isomorphic graphs among the four shown below.
SECTION 2.2 EULER CYCLES AND EULER PATHS

Euler cycles
A cycle which includes every edge of the graph (exactly once) is called an Euler cycle.
In other words, with an Euler cycle you can walk through the graph hitting every edge exactly once and end up back where you started.

criterion for the existence of an Euler cycle
Consider a connected graph (there's no hope for a non-connected graph).
It has an Euler cycle iff only if the degree of each vertex is even.

(1) If one or more vertex has odd degree then there is no Euler cycle.
(2) If all the vertices have even degree then there is an Euler cycle.

To see why (1) holds consider the graph in Fig 1 where vertex Z has degree 3. To get an Euler cycle you'd have to include edges e1, e2 and e3.
If you arrive at Z say on e1, leave on say e2, and eventually arrive again on e3 then you can't leave Z; there are no more exits so you're stuck and can't continue the cycle.
So a vertex of degree 3, or, more generally a vertex of odd degree, prevents Euler cycles.

(2) If all the vertices have even degree then there is an Euler cycle.

breakout algorithm
Suppose a graph is connected and all vertices have even degree. I'll illustrate how to find an Euler cycle using the graph in Fig 2.
Pick any vertex to start.
Begin traveling along successive edges, never repeating an edge.
Whenever a choice is available, pick the alphabetically or numerically first vertex to be systematic.
With this procedure, starting from vertex A in Fig 2 you get
A B C A stuck

You're stuck because all the edges at A have been used; there are no more exits available.
Note that you're stuck back at the initial vertex A again. You can't get stuck anywhere else because every vertex has even degree so there is an exit for every entrance.

So far you have the cycle ABCA (Fig 3) but it doesn't include all edges yet and there is no place to go from A. In this case, backtrack from the exit-less A until you find a vertex that does have exits, C in this case, and break out (there must be a way to break out to the edges not included yet since the graph is connected).

From C you have the side excursion C D B E C which starts at C and gets stuck back at C so you can pick up where you left off:
D B E C stuck
A B C A
Note that when you break out, the "remaining" degree at each vertex is even so on the side trips you always get stuck back where you broke out. This is what makes the algorithm work.

So far you have the cycle ABCDBECA in Fig 4 but not all edges are included yet so backtrack from the latest stuck position at C and find a place from which to break out, namely from E.

D F E stuck
D B E C
A B C A

Now there's nowhere to break out meaning that you have traversed all the edges and you're finished. An Euler cycle is

A B C D B E D F E C A (Fig 5)

example 1
You might get different Euler cycles by starting at different vertices. In Fig 2, starting from C you get

D F E
C A B C D B E C
↑

warning when you get back to C here, do not back up and break out because you are not stuck at C. There are still exits available from C. You only break out when you are back where you started and stuck.

This can be written neatly as

C A B C D B E D F E C

Starting from F you get F D B A C B E C D E F, no breakout necessary. The cycles starting from C and A respectively happen to be the same but the one you get from F is different. A graph can have many Euler cycles.

warning
To follow the spirit of this algorithm, remember to pick the first alphabetically of all vertices available.

Euler paths
An Euler path is a path which includes all edges (once each) but isn't a cycle (i.e., different starting and stopping points).

Let G be a connected graph. Then G has an Euler path iff G has precisely two vertices of odd degree (one to serve as the initial point with one more exit than entrance and one to serve as the end point with one more entrance than exit).
The breakout algorithm can be used to find an Euler path if you start at one of the vertices of odd degree. For example, look at the graph in Fig 6 where B and F have odd degree and the other vertices have even degree.

If you begin say with F you get

\[ \text{F I H D G H} \]
\[ \text{F C B A D E A} \]

so an Euler path is \text{F C B A D E F I H D G H E B}.

Note that when you start from one of the odd degree vertices you'll always get stuck at the other one: you can't get stuck at the initial vertex since it always has one more exit than entrance and you can't get stuck at one of the even-degree vertices since they always have as many exits as entrances.

And when you break out, the remaining degree at each vertex is even so on the side trips you always get stuck back where you broke out.

This makes the algorithm work.

**example 2**
The graph in Fig 7 has four vertices of odd degree \( \{A, B, C, D\} \).
So it doesn't have an Euler cycle or an Euler path.

**warning**
For small graphs it may be easy to find an Euler cycle by inspection. But the problems and the exams want you to use the breakout algorithm because it works automatically, even in large graphs that defy inspection.

**PROBLEMS FOR SECTION 2.2**

1. Each graph in the diagram is connected and all vertices have even degree. Find an Euler cycle starting from the starred vertex.
   (a)  
   (b)  
   (c)  
   (d)  

   (e)  
   (f)  

   (g)  
   (h)  

   (i)  
   (j)  

   (k)  
   (l)  

   (m)  
   (n)  

   (o)  
   (p)  

   (q)  
   (r)  

   (s)  
   (t)  

   (u)  
   (v)  

   (w)  
   (x)  

   (y)  
   (z)  

   (aa) 

   (ab) 

   (ac) 

   (ad) 

   (ae) 

   (af) 

   (ag) 

   (ah) 

   (ai) 

   (aj) 

   (ak) 

   (al) 

   (am) 

   (an) 

   (ao) 

   (ap) 

   (aq) 

   (ar) 

   (as) 

   (at) 

   (au) 

   (av) 

   (aw) 

   (ax) 

   (ay) 

   (az)
2. With the given floor plan (gaps represent doors), starting from outside the house, is it possible to walk through each interior and exterior door exactly once and end up outside again.

   Suggestion: Represent each room by a vertex, let the great outdoors (the "outside" room) be another vertex, Q, and draw an edge for each door between rooms.

3. Is it possible to draw the figure below without lifting the pencil from the paper and without retracing any lines. If not, explain why not. If possible, explain why you are sure it is possible and then do it.

4. Find an Euler cycle or an Euler path in the graph below if one exists.
SECTION 2.3 SPANNING TREES

definition of a tree
A graph which is connected and acyclic (no cycles) is called a tree. Fig 1 shows some trees and Fig 2 shows some non-trees.

![Fig 1 Three trees](image)

![Fig 2 Three non-trees](image)

spanning trees
Let $G$ be a connected graph. A spanning tree for $G$ is a tree, built with edges from $G$ and connecting all the vertices of $G$. Think of it as a skeleton for $G$, containing all of $G$'s vertices but usually with fewer edges (since $G$ may have cycles but a skeleton doesn't).

Every connected graph has a spanning tree (usually more than one). If $G$ is already a tree then $G$ is its own spanning tree; otherwise a spanning tree for $G$ has fewer edges than $G$. Fig 3 shows a connected graph (which is not a tree) along with several of its spanning trees.

A graph that isn't connected doesn't have a spanning tree.

![graph G](image)

![some spanning trees for G](image)

a spanning tree algorithm
There are two popular ways to construct a spanning tree for a connected graph starting from an arbitrary initial vertex.

breadth-first Add all the vertices directly connected by an edge to the initial vertex. Then for each of those newly added vertices in (alphabetical) turn, add all vertices directly connected that are not yet in the tree. And so on.

depth-first Add one vertex directly connected by an edge to the initial vertex. If there is more than one possibility, choose the first alphabetically to be systematic.

Then to that newly added vertex add one vertex (the first alphabetically) directly connected that is not yet in the tree. And so on.

If there is no way to grow the tree from the last vertex added, backtrack to the next-to-last and grow again. If necessary backtrack all the way to the initial vertex.
The breadth-first search grows a spanning tree by sending as many tentacles as possible out through the graph simultaneously. The depth-first search grows a spanning tree by sending one tentacle out through the graph.

I'll illustrate the algorithm by constructing some spanning trees for the graph in Fig 4.

example 1

Here's a breadth-first search starting from A in Fig 4.

round 1 Begin with A. Each of B, D, E is connected to A so add them to the tree (Fig 5 shows two versions of the spanning tree).

round 2 Look at B. Vertex C isn't in the tree yet and is connected to B so add C.

Look at D. Vertex G is not yet in the tree and is connected to D so add G.

Look at E. Vertices F and H are not yet in the tree and are connected to E so add them.

Now that all the vertices are in the tree, stop. Fig 5 shows two versions of the spanning tree.

example 2

Fig 6 shows the breadth-first tree (two versions) starting from E in Fig 4.

Fig 7 shows the breadth-first search starting from F in Fig 4.
example 3

Here's a depth-first search starting from A in Fig 4 (repeated below).

**round 1** Start at A. Each of B, D, E is connected to A. Pick one of them, say B, the first alphabetically, and add it to the tree (Fig 8).

Now start at B and look at all the vertices, C and E, connected to B but not in the tree yet. Pick one of them, say C, the first alphabetically, and add it to the tree.

Now start at C. None of the remaining vertices is connected to C so backtrack to B and look again.

**warning**

My version of the algorithm backtracks to the latest pick, B in this case, not A. Do it that way so that we'll all get the same answer.

**round 2** From B add E, then D and then G. You're stuck at G now so backtrack to E

**round 3** From E add F and then H and you're finished (Fig 8).

---

example 4

Fig 9 shows the depth-first search starting from E in Fig 4.

Fig 10 shows the depth-first search starting from G in Fig 4.

---

connection between the number of vertices and number of edges in a tree

A tree has one more vertex than edge, i.e.,

(1) \[ V = E + 1 \]

It follows that in a graph, if \( V \neq E+1 \) then the graph can't be a tree.

On the other hand, non-trees can have this property too so if a graph does have \( V = E + 1 \) the graph is not necessarily a tree. See problem 7(b).
why (1) holds

Look at the tree in Fig 11. Imagine growing it (Fig 12) with the depth-first spanning tree algorithm.

```
A
 • • •
B
•
C
• • •
D
•
E
•
F
```

FIG 11

```
A

•
B

C

D

E

F

2 vertices 3 vertices 4 vertices
1 edge 2 edges 3 edges
STEP 1 STEP 2 STEP 3 etc
```

Growing the tree in Fig 11

FIG 12

You start (Fig 12, step 1) with one less edge than vertex, namely with $V = 2$, $E = 1$. As the tree grows, at each stage, you join a new vertex to the growing tree, i.e., you add one more edge and one more vertex. So you continue to have one less edge than vertex. So that's what you have when the tree is fully grown.

example 5

Suppose a graph has 12 vertices and 15 edges. Then it can't be a tree because $V \neq E+1$.

Suppose a graph has 12 vertices and 11 edges. It may or may not be a tree.

PROBLEMS FOR SECTION 2.3

1. By inspection (forget fancy algorithms) decide how many spanning trees each graph has.
   (a) (b)

2. If a graph consists of a cycle with $n$ edges how many spanning trees does it have.

3. Find a spanning tree for the graph in the diagram.
   (a) breadth-first starting from A
   (b) breadth-first starting from D
   (c) depth-first starting from A
   (d) depth-first starting from D

4. Find a spanning tree for the graph in the diagram.
   (a) breadth-first from A
   (b) breadth-first from D
   (c) depth-first from A
   (d) depth-first from D
5. Here's the adjacency matrix for a connected graph. Without actually drawing the graph, find a spanning tree starting from v1
   (a) with depth-first (b) with breadth-first

   \[
   \begin{array}{ccccc}
   & v1 & v2 & v3 & v4 & v5 \\
   v1 & 0 & 1 & 0 & 1 & 0 \\
   v2 & 0 & 1 & 1 & 1 & 0 \\
   v3 & 0 & 0 & 0 & 0 & 0 \\
   v4 & 0 & 0 & 0 & 0 & 0 \\
   v5 & 0 & 0 & 0 & 0 & 0 \\
   \end{array}
   \]

   (matrix is symmetric)

6. Here are the adjacency matrices for two graphs. For each, without drawing a picture, use the spanning tree algorithm to decide if the graph is connected. If it isn't connected, how many components does it have?

   (a)

   \[
   \begin{array}{cccccc}
   A & B & C & D & E & F \\
   A & 0 & 0 & 1 & 1 & 0 \\
   B & 0 & 0 & 1 & 0 & 1 \\
   C & 1 & 1 & 0 & 0 & 0 \\
   D & 1 & 0 & 0 & 0 & 0 \\
   E & 1 & 1 & 0 & 0 & 0 \\
   F & 0 & 0 & 1 & 1 & 0 \\
   \end{array}
   \]

   (b)

   \[
   \begin{array}{cccccccc}
   A & B & C & D & E & F & G & H & I & J \\
   A & & & & & & & & & 1 \\
   B & & & & & & & & 1 & \\
   C & & 1 & & & & & & & \\
   D & 1 & & & & & & & & \\
   E & & 1 & & & & & & & \\
   F & & & & & & & 1 & & \\
   G & & & & & & & & & 1 \\
   H & & & & 1 & & 1 & & & \\
   I & & & & & 1 & & 1 & & \\
   J & & & & & & 1 & & & \\
   \end{array}
   \]

7. We know that the following statement is true.
   (*) If a graph is a tree then \( V = E + 1 \)

   (a) Find a counterexample to disprove the following statement.
   (**) If \( V = E + 1 \) then the graph is a tree.

   (b) True or False. And defend your answer.
   (***) If \( V \neq E + 1 \) then the graph is not a tree.

8. True or False
   (a) If \( G \) has 32 edges and 28 vertices then \( G \) is not a tree.
   (b) If \( G \) is not a tree then \( G \) is not connected and has at least one cycle.

9. If \( G \) is connected with 13 vertices and 17 edges, how many vertices and edges are there in a spanning tree for \( G \).
SECTION 2.4 PLANAR GRAPHS

definition of a planar graph

Fig 1 shows two edges crossing as opposed to Fig 2 where four edges have common endpoint T and no edges cross. A planar graph is one which can be drawn (redrawn if necessary) to not have crossing edges. The graph in Fig 2 is planar. The graph in Fig 1 is also planar because it can be redrawn (Fig 3) to avoid edges crossing. The graph in Fig 4 is planar because it can be redrawn so that no edges cross (Fig 4A).

showing that a graph is non-planar

I'll show that the graph in Fig 5 is not planar by showing (in a systematic fashion) that it can't be redrawn without edges crossing. Begin by drawing the largest circuit possible; Fig 6 shows one which includes all the vertices. I still have to put in edges AE, BF, CD. Suppose AE is drawn inside (Fig 7A). Then BF must go outside to avoid crossing AE (Fig 7B). But now CD can't be drawn without either crossing AE inside or FB outside. So Fig 5 is nonplanar.

warning

It is not convincing (especially on an exam) to try to show that a graph is non-planar by poking at its edges and simply declaring that they can't be redrawn without crossing. If you try to redraw, it must be with this systematic approach . An ad hoc approach may not be convincing to the reader.

faces of a planar graph

A planar graph divides the plane into disjoint regions called faces. One of them is the "outside" of the graph called the unbounded face or exterior face. The other faces are interior or bounded faces.

Fig 8 shows a planar graph with 4 faces; face IV is the unbounded face.

Fig 9 has 2 faces; II is the unbounded face.

If edge AD is added to Fig 9 as shown in Fig 10, the new graph still has 2 faces.

Euler's theorem

(1) In any planar connected graph, \( V - E + F = 2 \).

In other words, number of vertices - number of edges + number of faces = 2.
Note that the F in (1) includes the external face.
For the particular instance in Fig 11,
V = 5, E = 7, F = 4 (3 interior faces, 1 exterior face)
V - E + F = 5 - 7 + 4 = 2

why (1) holds
Let G be a planar connected graph with V vertices (Fig 12 gives an example).
Consider a spanning tree for G (Fig 13). The tree has V vertices, V - 1 edges and
one face (the unbounded face). So (1) holds for the spanning tree since in the tree,
\[ V - E + F = V - (V - 1) + 1 = 2 \]

Now add edges to the tree to get G back again. Each time you add an edge (Fig 14)
the number of vertices stays the same, the number of edges goes up by 1 and the
number of faces goes up by 1. So during the growth process, (1) continues to be true
and so it holds at the end of the process for G itself.

complete graphs
A graph which contains one edge between each pair of vertices is called complete.
The complete graph with n vertices is called \( K_n \) (Fig 15).

bipartite and complete bipartite graphs
A graph consisting of two sets of vertices so that edges go only from one set to
the other, with no edges within each set, is called bipartite (Fig 16).

A graph is complete bipartite if every vertex in one set is joined by one edge to
every vertex in the other set. In that case, if one set contains m vertices and the
other set contains n vertices, the graph is called \( K_{m,n} \) (same as \( K_{n,m} \) (Fig 17).
the famous nonplanar graphs \( K_{3,3}, K_5 \) and beyond

The graphs \( K_5 \) and \( K_{3,3} \) are nonplanar. Furthermore each is a dividing line between planar and nonplanar in the following sense:

- \( K_2, K_3, K_4 \) are planar; \( K_5, K_6, ... \) are non-planar.
- \( K_{m,n} \) is planar if \( m < 3 \) or \( n < 3 \) and nonplanar if both are \( \geq 3 \).

For example,

- \( K_{2,2}, K_{2,3}, K_{2,100} \) are planar.
- \( K_{3,3}, K_{3,4}, K_{4,4}, K_{4,100} \) are nonplanar.

why this holds

I already showed that \( K_{3,3} \) is nonplanar (go back to Fig 5 and the paragraph above it). Similarly (see problem 8) it can be shown that \( K_5 \) is nonplanar.

The "earlier" versions are planar: Fig 18 shows \( K_{m,2} \) redrawn without edges crossing. Fig 18A shows \( K_4 \) redrawn without edges crossing.

The later complete bipartite graphs, like \( K_{4,3}, K_{5,3}, ... \), and the later complete graphs, like \( K_6, K_7, ... \), are nonplanar because they contain \( K_{3,3} \) and \( K_5 \) respectively as a subgraph and any graph which contains a nonplanar subgraph must itself be nonplanar.

graphs homeomorphic to \( K_5 \) and \( K_{3,3} \)

Look at the graph in Fig 19. It is not \( K_5 \) because the "extra" vertex \( Q \) was added which splits edge \( BC \) into \( BQ \) and \( QC \). But Fig 19 is called homeomorphic to \( K_5 \). If \( Q \) is removed and edges \( BQ \) and \( QC \) merged into the single edge \( BC \) then it becomes \( K_5 \).

Similarly, Fig 20 is not \( K_{3,3} \) but it is homeomorphic to \( K_{3,3} \) because if the extra vertices \( P, S, R \) are removed and edges merged, it would become \( K_{3,3} \).
A graph is nonplanar if and only if it contains as a subgraph either $K_5$ or $K_{3,3}$ or something homeomorphic to one of them.

Part of the proof of Kuratowski's theorem is easy. Any graph which contains $K_5$ or $K_{3,3}$ or something homeomorphic to one of them is nonplanar because any graph which contains a nonplanar subgraph must itself be nonplanar. The other half of the proof is very long and is omitted.

warning Kuratowski's theorem does not say that every non-planar graph contains $K_{3,3}$ or $K_5$. (After all, the graph in Fig 19 is non-planar and contains neither $K_{3,3}$ nor $K_5$.) It says that every non-planar graph contains $K_{3,3}$ or $K_5$ or a homeomorphic copy of one of them.

example 1
The graph in Fig 22 is not $K_5$ (because of the multiple edges between B and Q). But it contains $K_5$ as a subgraph so it's nonplanar.
Fig 23 is not $K_5$ (because of Z). It doesn't contain $K_5$ as a subgraph. But it is homeomorphic to $K_5$ so it's nonplanar.

Fig 24 is not $K_5$ and does not contain $K_5$ as a subgraph. But it does contain a subgraph (the one in Fig 23) which is homeomorphic to $K_5$ so it's nonplanar.

**FIG 22**
Contains $K_5$

**FIG 23**
Homeomorphic to $K_5$

**FIG 24**
Contains a graph homeomorphic to $K_5$

**example 2**
Show that the graph in Fig 25 is nonplanar by finding a subgraph homeomorphic to $K_{3,3}$ or $K_5$.

The graph can't contain $K_5$ or something homeomorphic to $K_5$ since it doesn't have five vertices of degree 4 or more. So look for a buried $K_{3,3}$.

Vertex A is connected to B,F,E. So I'll look for two more vertices connected to B,F,E.

Vertex D can be connected to B via C, to F via I, and directly to E (Fig 26). Vertex H can be connected to B via J and G, directly to F, and to E through J (Fig 26). But using J twice makes it an extra vertex whose degree is more than 2 so Fig 26 is not homeomorphic to $K_{3,3}$.

The degree 2 requirement in the definition of homeomorphism means that in this search process a vertex can't be used as an extra more than once.

I tried re-routing H to B via C instead of J and G, leaving J for the H to E part. But then C is used twice. No good.

Then I gave up on H and tried J instead. Vertex J is connected to B via G, to F via H, and directly to E.

Fig 27 is homeomorphic to $K_{3,3}$ and it's a subgraph of Fig 25.

**FIG 25**

**FIG 26**
Not homeomorphic to $K_{3,3}$

**FIG 27**
homeomorphic to $K_{3,3}$

Note that Fig 27 is a subgraph of Fig 25 and with respect to the subgraph, the extra vertices G, H, I, C have degree 2 so the subgraph is homeomorphic to $K_{3,3}$. It's irrelevant that with respect to the original graph, G, H, I, C have degree 3.
the dual graph

If G is a planar connected graph then G has a dual graph G* constructed as follows. (1) In every face of G, including the unbounded face, place a vertex of G*. If the face is named P then it's convenient to name the vertex P also.

(2) If edge e in G is on the boundary between faces P and Q, draw edge e* in G* joining vertices P and Q.

In other words, faces in G correspond to vertices in G*, and vertices in G* are joined by an edge if the corresponding faces in G have a common boundary.

It's easier to understand with an example.

Fig 28 shows a planar connected graph; Fig 29 is its dual.

<table>
<thead>
<tr>
<th>original graph</th>
<th>dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>faces P, Q, R</td>
<td>vertices P, Q, R</td>
</tr>
<tr>
<td>edge 4 on boundary between faces P, Q</td>
<td>edge 4* joining vertices P, Q</td>
</tr>
<tr>
<td>edge 1 on boundary between faces P, R</td>
<td>edge 1* joining vertices P, R</td>
</tr>
<tr>
<td>edge 5 on boundary between faces Q, R</td>
<td>edge 5* joining vertices Q, R</td>
</tr>
</tbody>
</table>

**warning**

In Fig 28, edge 2 is a boundary between face P and the unbounded face R. Don't forget edge 2* in the dual. The dual should have an edge for each edge in the original.

Furthermore if you find the dual of G* the result will be the dotted graph in Fig 29, namely G. So G** = G.

In general, if G is a planar connected graph, then G and G* are duals of each other, i.e., G** = G.

Figs 30 and 31 show another graph and its dual. Edge 3 in Fig 30 is considered to be a boundary between face Q and itself and gives rise in G* to an edge 3* from vertex Q to itself, i.e., to a loop.
PROBLEMS FOR SECTION 2.4

1. Are these graphs planar or nonplanar?
   (a)  
   (b)  

2. The diagram shows a planar graph where the exterior face is bounded by the cycle ABHGA. Redraw it, still with no edges crossing, so that the exterior face is bounded by ABDCA.

3. Redraw the graphs in the diagram to show that they're planar and then (since they're also connected) verify Euler's theorem.
   (a)  
   (b)  

4. True or False
   (a) Adding loops can never change a graph from planar to nonplanar.
   (b) Adding multiple edges can never change a graph from planar to nonplanar.
   (c) Removing loops can never change a graph from nonplanar to planar.
   (d) Removing multiple edges can never change a graph from nonplanar to planar.

5. Verify Euler's theorem for a tree with 1000 vertices.

6. Suppose a planar connected graph has 9 vertices with degrees 2, 2, 2, 3, 3, 3, 4, 4, 5. How many edges? faces?

7. If a planar connected graph has 16 faces and all its vertices have degree 6 find the number of vertices.

8. Show that $K_5$ is non-planar by systematically trying to re-draw it.

9. Try to redraw each of the following (systematically) to see if it's nonplanar. For each nonplanar one, find a subgraph that is $K_{3,3}$ or $K_5$ or homeomorphic to one of them.
10. Show that the graph in the diagram is nonplanar by finding a subgraph which is $K_5$ or homeomorphic to $K_5$.

11. Show that if any edge is removed from $K_5$ the result is a planar graph
   (a) using Kuratowski's theorem
   (b) by actually redrawing the remainder with no edges crossing

12. What happens if you remove an edge from $K_6$. Is the result planar or non-planar.

13. Draw the dual graph.
   (a)

14. Given the dual in the diagram, draw the original.

15. (a) Draw the dual of the tree in the diagram.
    (b) Recopy the dual to get a fresh start and draw the dual of the dual to check you get the original back again.
REVIEW PROBLEMS FOR CHAPTER 2

1. Find a spanning tree for the graph in the diagram breadth-first from F

```
A  B  C  D
E   F  G
I   J  K
M  N  P
```

2. Suppose a spanning tree for a connected planar graph G consists of edges $e_1, e_2, \ldots, e_n$
   True or False and defend your answer.
   The edges $e_1^*, e_2^*, \ldots, e_n^*$ form a spanning tree for the dual $G^*$.

3. Are these graphs isomorphic? Explain.

```
A B
C D
```
```
S U
T P
```

4. Find an Euler cycle if possible starting from vertex E. If not possible explain why not.

```
R
Q
C
F
```
```
B
A
E
```

5. I have a connected graph G with 27 vertices and 50 edges.
   Fill in each blank with MUST, MIGHT, or CAN'T.
   (a) If I throw away 25 edges, the resulting graph _____ be a spanning tree for G.
   (b) If I throw away 24 edges, the resulting graph _____ be a spanning tree for G.
   (c) If I throw away one vertex and 25 edges, the resulting graph _______ be a spanning tree for G.

6. Here are two isomorphic graphs.
   (a) Draw their duals
   (b) Show that their duals are not isomorphic.
   (c) Does that bother you? If the originals are really the "same" shouldn't they have the same duals.
7. If a planar connected graph has 16 edges and each vertex has degree 4, how many faces does it have.

8. A planar connected graph G is called self-dual if G and G* are isomorphic. Is the graph in the diagram self-dual.

9. Show that the graph in the diagram (a cube plus 4 diagonals) is nonplanar by finding a subgraph which is $K_{3,3}$ or homeomorphic to $K_{3,3}$. 