Question set for Chapter 2

1. Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are connected graphs. Let x be a vertex of G_1 and let y be a vertex of G_2. Form a new graph $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{xy\})$.

   (a) Draw the graph G if G_1 and G_2 are specified by:

   $$V_1 = \{a, b, c\},\ E_1 = \{ab, bc, bc\},\ V_2 = \{d, e, f, g\},\ E_2 = \{df, dg, dg, eg, eg\},$$

   and if $x = c$ and $y = g$.

   (b) Prove that the new graph G is connected (i.e. not just the one you constructed in part (a) but *any* G you construct in this way).

2. It is required to design an algorithm

   st_Path(G,s,t)
   Input: graph G, vertices $s, t \in V(G)$
   Output: True if there is a path from s to t in G; False otherwise

   (a) Specify suitable pseudocode for st_Path.
   [Hint: adapt the pseudocode for Maximal_Subtree given in the lecture notes. You will **NOT** be asked to answer a question like this in the final exam, but it is good practice for thinking about algorithms.]

   (b) Show how your algorithm runs on the following input, showing **diagrammatically** each step which the algorithm makes:

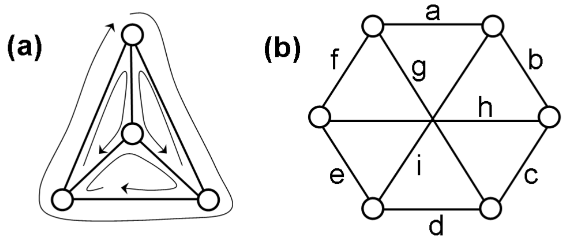
   $$\begin{aligned} V(G) &= \{a, b, c, d, e, f, g, h\}; \\ E(G) &= \{ab,\ ac,\ ad,\ cd,\ de,\ ef,\ eg,\ eh,\ fg\}; \\ s &= a,\ t\ =\ g. \end{aligned}$$

3. A **bridge** in a connected graph G is an edge whose deletion will disconnect the graph. One of the most famous open questions in graph theory, posed independently in the 1970's by Paul Seymour and George Szekeres, is the following:

   The Cycle Double Cover Conjecture If G is a graph with no bridges then there is a list of cycles in G such that every edge of G lies in exactly two of these cycles.

   For example in the graph (a) on the right (which is called K_4) four cycles may be chosen as shown.

   Specify a list of cycles which provides a cycle double cover for the graph in (b).



4. Run the Maximal_Subtree algorithm on the graph shown on the right, starting at
   (i) vertex a; (ii) vertex b;
   showing each iteration your algorithm makes to construct a maximal subtree. In each case, give the number of vertices and edges contained in the output subtree and state whether or not the subtree is maximum.