1. {6 marks}

   (a) {4 marks}Let $S_{n,k}$ be the subset graph as defined in the course notes in Example 4.1.5 (see course notes figure 4.5 for an example). Prove that for $n \geq 3$, $k \geq 2$, and $n > k$, $S_{n,k}$ is not bipartite.

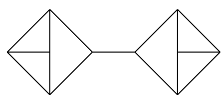
   Solution. Consider the three vertices $\{1, 2, \ldots, k\}$, $\{1, 2, \ldots, k-1, k+1\}$, and $\{2, 3, \ldots, k, k+1\}$. These are well defined and distinct since $n > k$ and $k \geq 2$. Any pair of them shares exactly $k-1$ elements. Therefore these three vertices form a cycle of lenght 3 in $S_{n,k}$. So $S_{n,k}$ contains an odd cycle and hence, by a result from class, is not bipartite.

   (b) {2 marks}True or False: every bipartite graph contains an even cycle. If the statement is true give a proof and if the statement is false give a counterexample.

   Solution. This statement is false. Any forest is a counterexample (the question doesn't say connected, so forests which aren't trees are ok, but trees are also ok). To keep it simple, a specific counterexample is the path of length 1 on two vertices. This is bipartite with each vertex in its own part of the bipartition and it has no cycles at all so in particular no even cycles.

2. {6 marks}

   (a) {2 marks} Draw a 3-regular graph with at least one bridge.

   Solution. Here's one 

   (b) {4 marks} Prove that a 4-regular graph can not contain a bridge.

   Solution. Suppose there were a 4-regular graph G with a bridge e. Let u and v be the ends of e. Let H be the connected component of G containing e. Then H is also a 4-regular graph with a bridge since e is also a bridge of H. By the properties of bridges, $H - e$ has 2 connected components one containing u and the other containing v. Consider the connected component of $H - e$ containing u. In this connected component u has degree 3 and all other vertices have degree 4 since G was 4-regular. However every graph has an even number of vertices of odd degree, so this is a contradiction. Therefore no 4-regular graph with a bridge exists and so a 4-regular graph can not contain a bridge.

3. {6 marks}

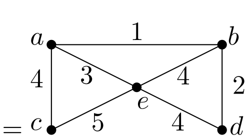
   (a) {4 marks} Let C_n be the n-cube as defined in the course notes in Example 4.4.1 (see course notes figure 4.12 for an example). Prove that C_n does not have any bridges for $n \geq 2$.

   Solution. By a result from class, to show C_n does not have any bridges it suffices to show that every edge of C_n is in a cycle. Let e be an edge of C_n. Then the ends of e are two binary strings of length n which differ in exactly one bit. Say the strings are $s_1 = w_1 w_2 \cdots w_i \cdots w_n$ and $s_2 = w_1 w_2 \cdots w_i' \cdots w_n$ where the w_k are bits and w_i and w_i' are opposite bits (that is $w_i' = 1 - w_i$). Since $n \geq 2$ we can pick $j \neq i$, $1 \leq j \leq n$. Let w_j' be the opposite bit from w_j ($w_j' = 1 - w_j$). Consider $s_3 = w_1 w_2 \cdots w_j' \cdots w_n$ and the binary string s_4 with both w_j' in place of w_j and w_i' in place of w_i'. Both s_3 and s_4 are also vertices of C_n and s_1, s_2, s_4, s_3, s_1 determines a cycle of C_n containing the edge e. Therefore C_n has no bridges.

   (b) {2 marks} Suppose you have a graph G, a vertex v of G, and a spanning tree T of G. True or false: you can choose weights for the edges of G so that running Prim's algorithm on G beginning at vertex v gives T. If true describe the weights and explain why they work, if false give an example for which you prove that it is impossible.

   Solution. This statement is true. Suppose I set the weights of all the edges of T to be 1 and the weights of all the other edges to be 2. Then at each iteration of the loop since I do not have all of T yet but T is connected there will be at least one edge of T which can be chosen, and one such edge will be chosen since the edges of T all have strictly lower weight than the other edges. Thus at each iteration of the loop we will pick one additional edge of T and the loop will end when we have picked all of T.

4. {5 marks} Find a minimal spanning tree for the graph G using Prim's algorithm starting at vertex a.

$G =$ 

Show both

- the spanning tree you construct on a drawing of the graph and

- a table showing which edges and vertices you add to the tree at each iteration of the loop.

Solution.

loop iteration	vertices of tree so far	edges of tree so far
before loop	$\{a\}$	$\emptyset$
1	$\{a, b\}$	$\{ab\}$
2	$\{a, b, d\}$	$\{ab, bd\}$
3	$\{a, b, d, e\}$	$\{ab, bd, ae\}$
4	$\{a, b, c, d, e\}$	$\{ab, bd, ae, ac\}$

So the spanning tree we construct is