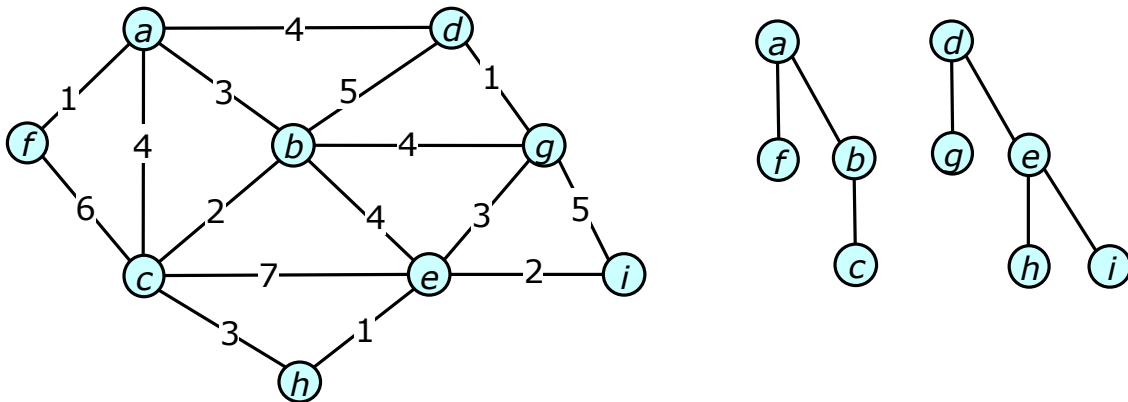# Practice Questions for Exam 2

1. (10 points) The diagram below shows a graph and the partition data structure from some intermediate point in the execution of Kruskal's algorithm.

Which edges in the graph are in the partial MST solution computed so far?

In the partition data structure, what is the value of *rank(a)*?

What is the value of *rank(e)*?

Consider an alternate version of Kruskal's algorithm that does not use the partition data structure. Instead, when considering an edge {*u,v*} it does a tree traversal in the current "blue tree" containing vertex *u* and checks to see if *v* is in the tree. Give a big-*O* bound on the running time of this version of Kruskal's algorithm and explain why the running time can be as bad as your bound.

2.   (15 points) Consider a find operation on the partition data structure in which the "find path" has 21 nodes. What is the smallest number of nodes that the partition data structure could contain. Explain your answer.

In the $O(m \log \log n)$ analysis of the partition data structure, we defined the notion of a *dominant* node. Suppose the partition data structure has 16 million nodes. If the root of the tree involved in the above find operation has a rank of 30, what is the maximum number of nodes along the find path that could be dominant? Explain your answer.

Consider a node x with $rank(x)=0$ and $rank(p(x))=8$. Suppose we perform three operations involving node x and x is not dominant immediately before each of these find operations. What is the smallest possible value that $rank(p(x))$ can have after these three find operations have completed? Explain your answer.

3. (12 points) The diagram below shows a partial representation of the state of the round-robin algorithm. Specifically, it shows the graph, the tree edges and the leftist heaps (with keys and ranks omitted).
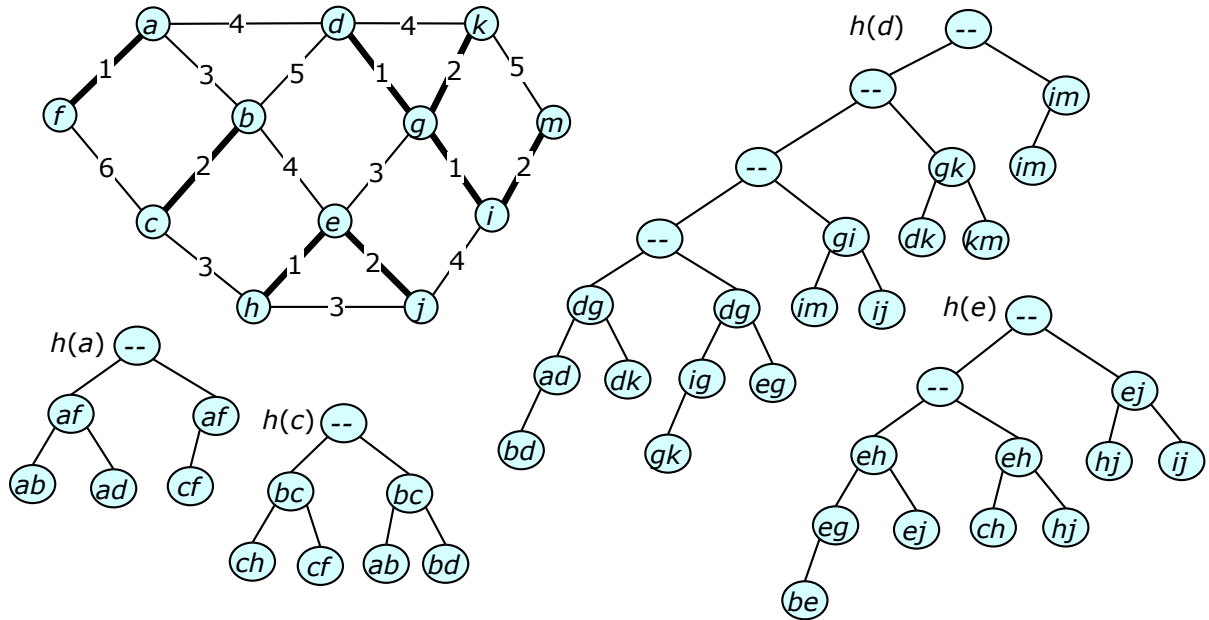

List the sets defined by the partition data structure for this state. For each set in the partition, circle the canonical element.

In the diagram above draw an X through all heap nodes in $h(e)$ that are considered deleted by the algorithm.

If the tree containing e is at the front of the round-robin queue at this point in time. List the subtrees of $h(e)$ that are returned by the *purge* method in the *findmin* operation at the start of the main loop. Also, show the heap that is returned by the *heapify* method, within *findmin*.

4. (15 points) The figure below shows an intermediate stage in the execution of Edmonds
   algorithm for finding a maximum size matching. The matching edges are not shown
   explicitly, but parent pointers are shown and the partition data structure is shown.
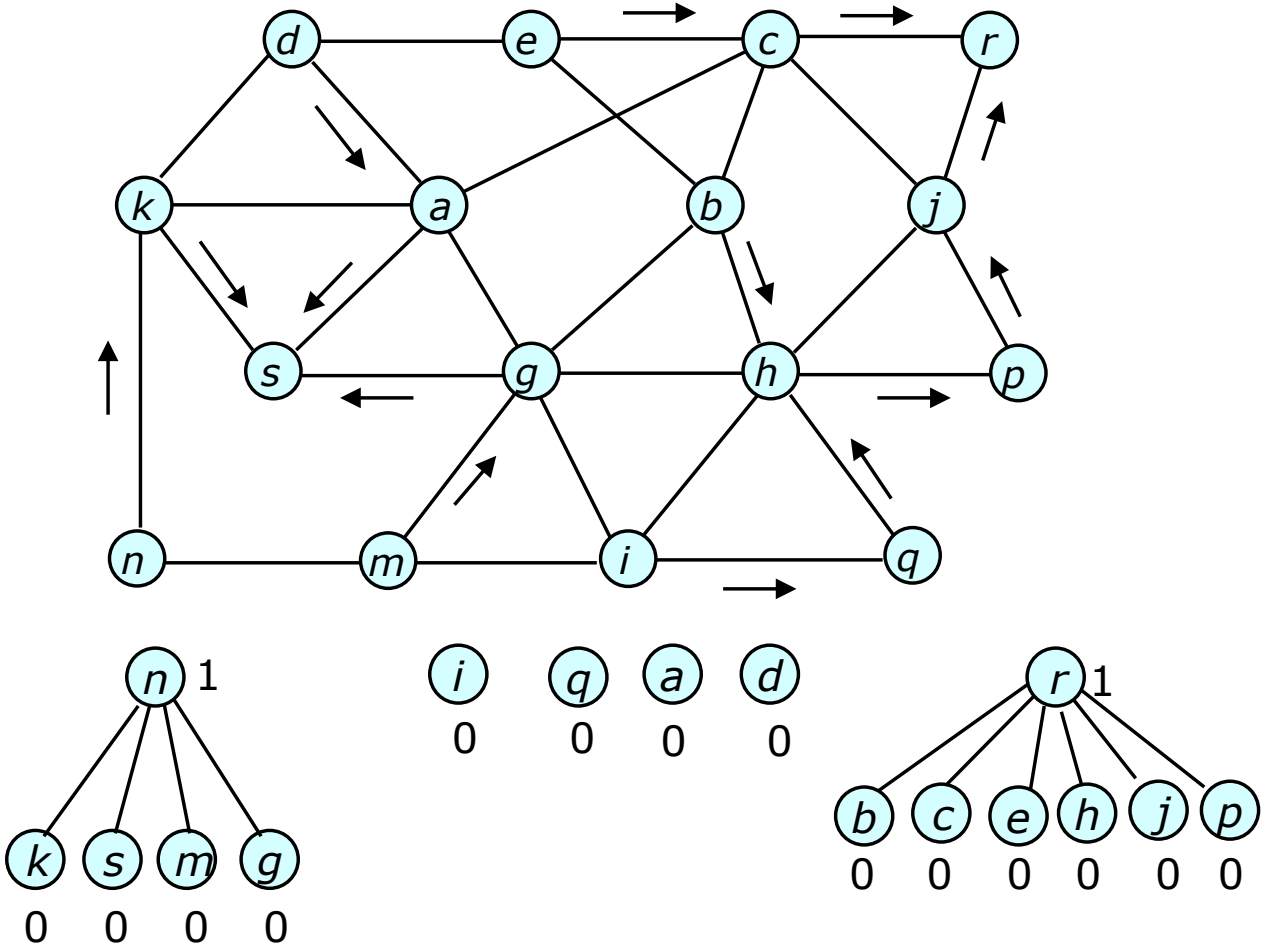
Draw closed curves around the vertex sets which form blossoms in the current graph.

Mark the bridges of the blossoms with a *B*.

Mark even vertices with a plus sign and odd vertices with a minus sign.

Indicate which edges are in the current matching, by marking them with an *M*.

If edge {*g,h*} is processed next, an augmenting path is found. List the vertices in that
augmenting path.

5. The graph at right is an instance of the symmetric traveling salesman problem with distances that satisfy the triangle inequality. Show how to construct an optimal solution using Christofides' algorithm. First, mark the edges in a minimum spanning tree.
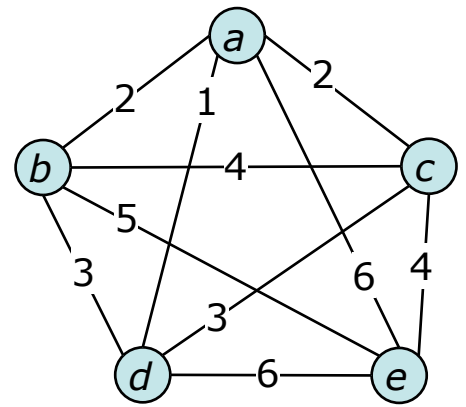
Next, draw a diagram of the "odd-degree subgraph" with edge weights and mark a subset of the edges that correspond to a minimum weight perfect matching.

List the edges in the Eulerian tour, based on your answers to the previous parts.

List the edges in the resulting TSP tour.

Consider another instance of TSP and a solution to Christofides' algorithm. Suppose the MST has a cost of 17, the min weight perfect matching has a cost of 12 and the TSP tour produced by the algorithm has a cost of 26. Estimate how close this tour is to the optimal tour. Justify your answer.

6. Consider the LP instance shown below.

$$\text{maximize} \begin{bmatrix} 2 & 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{subject to} \begin{bmatrix} 1 & 4 & -1 & 2 \\ 2 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \leq \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

Write the corresponding dual in matrix form.

Let $X=[1\ 2\ 6\ 1]^T$ and $Z=[3\ 7]^T$ be feasible solutions to the primal and dual problems. Write out the corresponding slack vectors.

Do these satisfy the complementary slackness conditions.

Consider another LP instance with primal and dual solutions X and Z. Suppose that for the primal, the value of the objective function with respect to X is 17 and for the dual, the value of the objective function is also 17. Do these solutions satisfy the complementary slackness conditions? Explain.

7. The diagram below shows an intermediate state in the execution of Edmond's algorithm for max weight matchings in bipartite graphs. The numbers next the vertices are the labels.
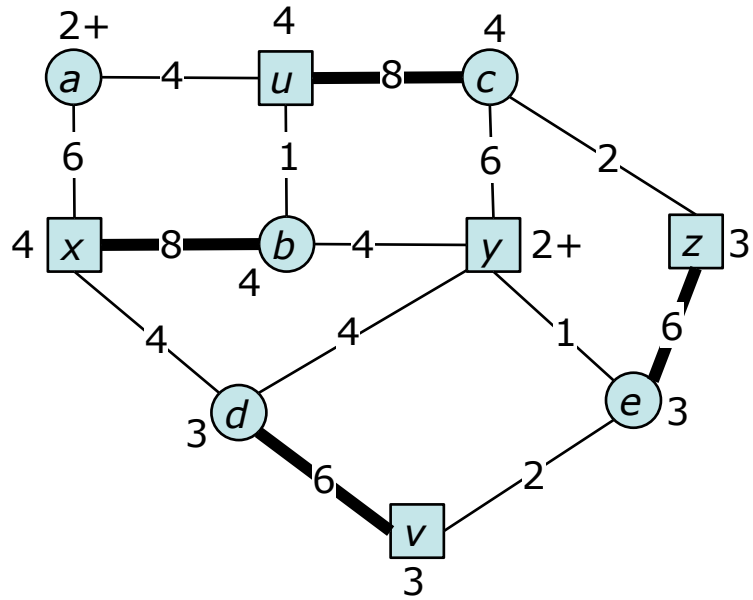
If the algorithm proceeds from this point, it builds a set of trees. Show these trees on the diagram using arrows to indicate parent pointers and plus/minus signs to indicate the state of vertices in the collection of trees.

Once the algorithm builds the set of trees, it stops to relabel the vertices. What are the values it computes for δ_1, δ_2 and δ_3 in this step?

What are the labels for vertices b, c and d after this step takes place?

The relabeling creates some new equality edges. What are the new equality edges?

What does the algorithm do next?

8. The diagram at right is a *partial* representation of the state of Edmonds algorithm for finding max weight matchings in general graphs. The numbers next to the vertices are their labels. The labels for the two blossoms are also shown.
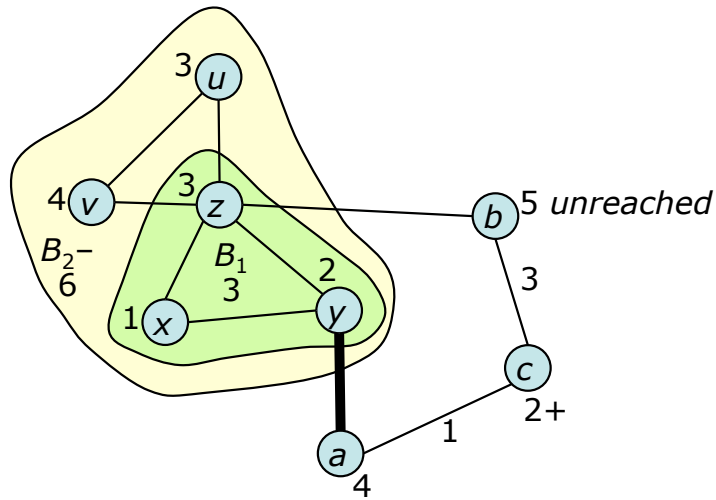
Which of the edges in the blossoms belong to the matching?

What are the states of vertices a and u?

What are the values of $w(u,z)$, $w(a,y)$, $w(x,y)$?

If a relabeling step is done next, what are the values of δ_1, δ_2, δ_3 and δ_4 (assuming no other information)?

What are the new labels assigned to u, a, x, B_1 and B_2?

What would the algorithm do next?