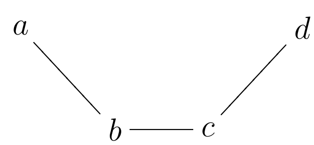
Homework 5

0. Problems from sections 8 and 9:

8.5.3. A connected component with k nodes has at least $k-1$ edges because a tree is the minimal connected graph on k nodes and trees always have $k-1$ edges. Consider a graph with p connected components and $k_1, k_2, \ldots, k_p$ vertices in the respective components. Since every vertex lies in a unique connected component, the graph has $n = k_1 + \ldots + k_m$ vertices and at least $n-p$ edges. Hence, the number of edges $m \geq n-p$ and so $p \geq n-m$ as desired.

8.5.5. We can use the tree-growing algorithm to generate all such trees, and then determine which trees are isomorphic. The number of trees for $n = 1, \ldots, 6$ turns out to be 1, 1, 1, 2, 3, 6. Sloane's has more on this sequence at `http://www.research.att.com/~njas/sequences/A000055`.

8.5.6. Every double star has a structure in which there exists a subgraph



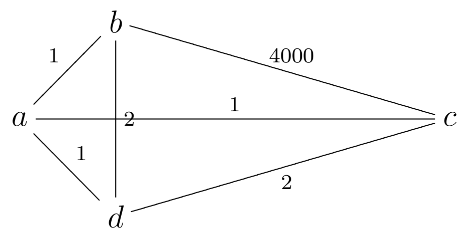
and each of the other nodes are attached to either b or c. Hence, each double star will be determined if we know the degrees of b and c. Since the degree of c is $n - deg(b)$, and interchanging the vertices b and c gives the same double star, suppose we choose the degree of b so that $deg(b) \leq deg(c)$. Then, the degree of b can be any of $\{2, 3, \ldots, 2 + \lfloor \frac{n-4}{2} \rfloor\}$. Hence, there are $\lceil \frac{n-4}{2} \rceil$ double stars.

9.2.3. Suppose all of the edge costs are distinct. Recall the proof that Kruskal's algorithm gives a lowest-cost tree: Given any tree G, we produce a tree H with lower cost that agrees with the greedy tree F on more edges. Since H is constructed by removing an edge from G and replacing it with an edge from the greedy tree, we actually have the *strict* inequality $cost(H) < cost(G)$, because the edge costs are all distinct. Replacing G by H and running the argument again produces a chain of *strict* inequalities

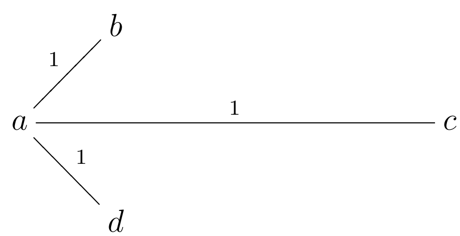
$$cost(F) = cost(H_k) < \cdots < cost(H_2) < cost(H_1) < cost(G).$$

Therefore, every tree $G \neq F$ is strictly more expensive than F, so F is the unique cheapest tree.

9.2.8. Consider the graph



Then, the unique lowest cost spanning tree is



If we start at d and apply the Tree Shortcut Algorithm, we will traverse the nodes in order d, a, b, a, c, a, d, and after applying shortcuts, we obtain d, a, b, c, d as our output tour. This has cost 4004, which is more than 1000 times the optimal tour d, c, a, b, d with cost 4.

1. Prove that if G is connected and $|E| = |V| - 1$ then G is a tree. Also, prove that if G is cycle-free and $|E| = |V| - 1$ then G is a tree.

Suppose $G = (V, E)$ is connected, but G is not a tree. Then, G must contain cycles. We can find a spanning tree $T = (V, E')$ of G with the same nodes as G but strictly fewer edges. Hence, $|E| > |E'| = |V| - 1$, contradicting that $|E| = |V| - 1$.

Next, suppose that $G = (V, E)$ is cycle-free, but G is not a tree. Then G must be disconnected. We can add edges to obtain a spanning tree $T = (V, E')$ with the same nodes as G but strictly greater edges. Hence, $|E| < |E'| = |V| - 1$, contradicting that $|E| = |V| - 1$.