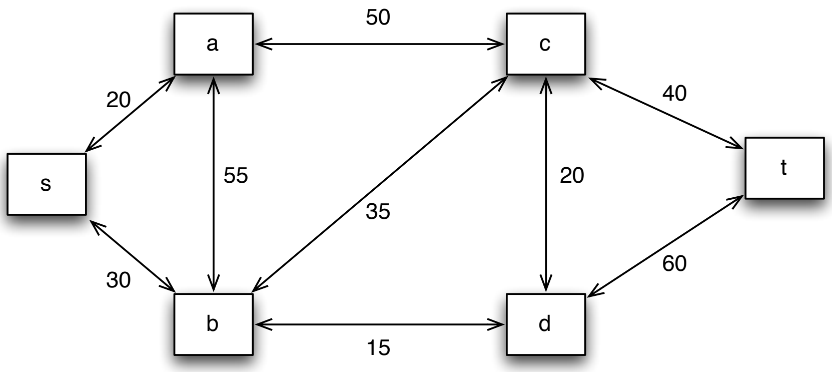
Homework 2

Vancouver Summer Program 2017
Algorithms and the Internet
Due: August 2, 2017, by 11:59 p.m.

---

1. (Applying graph algorithms.)

   (a) Find the shortest path between s and t in the following graph.

   (b) The faculty members in ECE are assigned to different committees (e.g., curriculum, student affairs, safety, operations). A faculty member may be part of more than one committee. Each committee meets on a weekly basis for an hour. You are given information about the membership of each committee and you need to decide on the number of meeting rooms to book and the time slots to use for the meetings. It is preferred that as many meetings as possible happen concurrently on Monday mornings so that committee members can work on action items during the week. Naturally, if someone is on two committees, the two meetings cannot be held concurrently. Describe how the problem of scheduling meetings can be mapped to a graph colouring problem. You must clearly describe what vertices and edges represent in the graph construction.

   You do not need to write an algorithm. You want to describe a graph that represents this scheduling problem and show that colouring the graph correctly would solve the problem.

2. (Some simple graph properties.) Let G be a graph with v vertices and e edges. Let M be the maximum degree of the vertices of G, and let m be the minimum degree of the vertices of G. Which of the following propositions must be true? Provide a short proof or counterexample in each case.

   (a) $2e/v \leq M$

   (b) $2e/v \geq m$

   (c) There exists a simple path (includes no cycles) of length at least m

(d) $m > 2$ implies that G is connected

(e) $M \leq v - 1$ if G is a simple graph

3. (Some more graph properties.) Let m be a positive integer and consider a graph G^* with $2m$ vertices: $v_1, \ldots, v_{2m}$. An edge exists between vertices v_i and v_j if and only if $(i - j \equiv 1 \mod 2m) \vee (i - j \equiv 2m - 1 \mod 2m) \vee (i - j \equiv m \mod 2m)$.

   Note that $x \equiv y \mod n$ if and only if $x = kn + y$ for some integer k. As examples, $25 \equiv 5 \mod 20$, $29 \equiv -1 \mod 30$ and $29 \equiv 29 \mod 30$.

   (a) For each $j \in \{2, \ldots, 2m\}$, what is the distance between v_1 and v_j? The *distance* between two vertices of a graph is the number of edges on the shortest path that connects the two vertices. (Derive an expression in terms of i, j and m. You will have to consider a few cases.)

   (b) A graph G is k-edge-connected if and only if one has to remove k edges to disconnect the graph. Prove that G^* is not 4-edge-connected: you can remove three or fewer edges to disconnect the graph.

4. (Paths in a graph.) Show that in every (simple) graph there is a path from any vertex of odd degree to some other vertex of odd degree.

5. (Directed acyclic graphs). A directed graph which has no directed cycles is called a directed acyclic graph (DAG). Note that the underlying undirected graph may have cycles.

   (a) Show that in any DAG, there is a vertex whose in-degree equals 0, i.e., it is not the head for any edge. Such a vertex is called a *source*.

   (b) Show that in any DAG, there is a vertex whose out-degree equals 0, i.e., it is not the tail for any edge. Such a vertex is called a *sink*.

   (c) Show that in any DAG, one can order the vertices so as to respect edge directions: i.e., show there exists a one-to-one and onto mapping $f : V(G) \to \{1, \ldots, n\}$ such that for every directed edge (u, v), $f(u) \leq f(v)$. So every edge points from a lower numbered vertex to a higher numbered vertex. This kind of an ordering is called a *topological sort* of the vertices of a DAG.

   (d) We say that a partition of the vertices $V = L_0 \cup \cdots \cup L_{k-1}$, where $L_i \cap L_j = \emptyset$ for all $i \neq j$, is a *stratification* with k levels, if every directed edge e is between vertices in different levels, and the edge points from a lower indexed level to a higher indexed level. Notice that the topological sorting from the previous part is a stratification with $|V|$ levels. Let G_0 be a DAG and let L_0 be the set of sources in G_0. Consider the following graphs. For each $i = 1, 2, \ldots,$ define $G_i = G_{i-1} \setminus L_{i-1}$ and L_i is the set of sources in G_i. Notice that for some natural number k^*, for every $i \geq k^*$, G_i is just the empty graph, i.e., empty set of vertices and empty set of edges. Show that $V = L_0 \cup L_1 \cup \cdots L_{k^*-1}$ is a stratification.

   (e) (**Bonus**) Show that k^* in the previous part is the smallest $k \in \mathbb{N}$ such that there exists a stratification with k levels.

6. (BFS–Proof of Correctness) Consider Breadth First Search (BFS) running on a connected graph G and starting at some source vertex $x \in V$. We claimed that BFS grows a tree as it traverses the graph, and that after BFS terminates, it will have

constructed a shortest path (to x) tree. Consider T in procedure BFS below (Algorithm 1). (Note: you may add lines to the BFS procedure below if this would aid in proving any of the statements.)

(a) Show that, after BFS terminates, T is a connected subgraph of T.

(b) Show that T spans G.

(c) Show that T is a tree.

(d) Show that for every $y \in V(T)$, the path from x to y in T is a shortest path from x to y in G.

Algorithm 1 BFS(G, x)

```
Q ← [x], T ← {x}
while Q is not empty do
    Remove a vertex v from front of Q
    visit v
    for each unmarked neighbor w of v do
        Mark w
        Add w to end of Q
        Add edge vw to T
    end for
end while
```