Staff Solutions to In-Class Problems Week 7, Mon.

**STAFF NOTE:** Digraphs: Walks & Paths, Ch. 9–9.4

**Problem 1.** (a) Give an example of a digraph in which a vertex $v$ is on a positive even-length closed walk, but no vertex is on an even-length cycle.

**STAFF NOTE:** Hint: There is an example with three vertices.

**Solution.** A graph consisting of three vertices in a cycle is an example. That is, the vertices are $a, b, c$ and the edges are $\langle a \to b \rangle, \langle b \to c \rangle, \langle c \to a \rangle$. The only cycle in the graph is of length three, but of course going around it twice gives an even-length closed walk.

(b) Give an example of a digraph in which a vertex $v$ is on an odd-length closed walk but not on an odd-length cycle.

**Solution.**

$$V ::= \{a, v, c\},$$

$$E ::= \langle a \to a \rangle, \langle a \to v \rangle, \langle v \to a \rangle.$$ 

Now $v \langle v \to a \rangle a \langle a \to v \rangle$ is a length three closed walk, but there is only one cycle that includes $v$, and it has length two.

(c) Prove that every odd-length closed walk contains a vertex that is on an odd-length cycle.

**Solution.** Proof. Suppose to the contrary that there was an odd-length closed walk that did not contain a vertex that was on an odd-length cycle. Let $e$ be a shortest such walk. Now $e$ cannot itself be a cycle or all its vertices would be on an odd-length cycle. So $e$ must have an additional repeat vertex besides its beginning and end. There are then two cases to consider depending on whether the additional repeat is different from, or the same as, the start vertex.

**Case 1.**

$$e = a f \hat{b} g \hat{b} h a$$

for some vertices $a \neq b$ and positive length walks $f, g, h$.

Now if $g$ has odd length, then, being shorter than $e$, it contains a vertex on an odd-length cycle. This contradicts our hypothesis that $e$ does not contain such a vertex.

So $g$ must have positive even length, which implies that

$$a f \hat{b} h a$$


© 2015, Albert R Meyer. This work is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 license.
is an odd-length closed walk, which being shorter than \( e \), contains a vertex on an odd-length cycle, again contradicting our hypothesis that \( e \) does not contain such a vertex.

**Case 2.**

\[ e = a \vec{f} \vec{a} g a. \]

for positive length closed walks \( f \) and \( g \). Since the length of \( e \) is odd and equal to the sum of the lengths of \( f \) and \( g \), at least one of \( f \) and \( g \) must be an odd-length closed walk, which being shorter than \( e \), contains a vertex on an odd-length cycle, again contradicting our hypothesis that \( e \) does not contain such a vertex. 

---

**Problem 2.**

Lemma 9.2.5 states that \( \text{dist}(u, v) \leq \text{dist}(u, x) + \text{dist}(x, v) \). It also states that equality holds iff \( x \) is on a shortest path from \( u \) to \( v \).

(a) Prove the “iff” statement from left to right.

**Solution.** *Proof.* To prove the “iff” from left to right, suppose \( \text{dist}(u, v) = \text{dist}(u, x) + \text{dist}(x, v) \). Then merging a shortest path from \( u \) to \( x \) with shortest path from \( x \) to \( v \) yields a walk whose length is \( \text{dist}(u, x) + \text{dist}(x, v) \), which by assumption equals \( \text{dist}(u, v) \). This walk must be a path or it could be shortened, giving a smaller distance from \( u \) to \( v \). So this is a shortest path containing \( x \). 

(b) Prove the “iff” from right to left.

**Solution.** *Proof.* To prove the “iff” from right to left, suppose vertex \( x \) is on a shortest path \( w \) from \( u \) to \( v \), namely, \( w \) is a shortest path of the form \( f \vec{x} r \). The path \( f \) must be a shortest path from \( u \) to \( x \); otherwise replacing \( f \) by a shorter path from \( u \) to \( x \) would yield a shorter path from \( u \) to \( v \) than \( w \). Likewise \( r \) must be a shortest path from \( x \) to \( v \). So \( \text{dist}(u, v) = |w| = |f| + |r| = \text{dist}(u, x) + \text{dist}(x, v) \). 

---

**Problem 3.**

A 3-bit string is a string made up of 3 characters, each a 0 or a 1. Suppose you’d like to write out, in one string, all eight of the 3-bit strings in any convenient order. For example, if you wrote out the 3-bit strings in the usual order starting with 000 001 010 . . . , you could concatenate them together to get a length \( 3 \cdot 8 = 24 \) string that started 000001010 . . . .

But you can get a shorter string containing all eight 3-bit strings by starting with 00010 . . . . Now 000 is present as bits 1 through 3, and 001 is present as bits 2 through 4, and 010 is present as bits 3 through 5, . . . .

(a) Say a string is 3-good if it contains every 3-bit string as 3 consecutive bits somewhere in it. Find a 3-good string of length 10, and explain why this is the minimum length for any string that is 3-good.

**Solution.** The string 0001110100 is a length 10 string that is 3-good. You can’t do better: there must be two bits to start and each additional bit can yield at most one new 3-bit string. 

(b) Explain how any walk that includes every edge in the graph shown in Figure 1 determines a string that is 3-good. Find the walk in this graph that determines your 3-good string from part (a).

**Solution.** A string can be built up from any walk by starting with the \( k \) bits in the vertex at the start of the walk and successively adding the bit that labels the edge to the end of the string being built. If the walk includes every edge, then any string \( b_1b_2b_3 \) will appear as a substring when the edge \( (b_1b_2 \rightarrow b_2b_3) \) appears in the walk.
In particular, the string 0001110100 is determined by the walk that goes through the following sequence of edges:
\[
(00 \rightarrow 00) \ (00 \rightarrow 01) \ (01 \rightarrow 11) \ (11 \rightarrow 10) \ (10 \rightarrow 01) \ (01 \rightarrow 10) \ (10 \rightarrow 00).
\]

(c) Explain why a walk in the graph of Figure 1 that includes every edge exactly once provides a minimum-length 3-good string.1

**Solution.** Since there are 8 edges, the string determined by the walk will be of length 10, which is the minimum possible as observed in part (a). Since the walk includes every edge, it will determine a 3-good string by part (b).

(d) Generalize the 2-bit graph to a \(k\)-bit digraph, \(B_k\), for \(k \geq 2\), where \(V(B_k) := \{0, 1\}^k\), and any walk through \(B_k\) that contains every edge exactly once determines a minimum length \((k + 1)\)-good bit-string.2 What is this minimum length?

Define the transitions of \(B_k\). Verify that the in-degree and out-degree of every vertex is even, and that there is a positive path from any vertex to any other vertex (including itself) of length at most \(k\).

**Solution.** \(2^{k+1} + k\).

A bit-string of length \(n\) has exactly \(n - k\) locations where a length \(k + 1\) subsequence can begin. Since there are \(2^{k+1}\) length-(\(k + 1\)) bit-strings, the minimum length, \(n\), of any \((k + 1)\)-good bit-string must satisfy \(n - k \geq 2^{k+1}\), so the minimum length is \(2^{k+1} + k\). This is exactly the length of the bit-string that would be determined by a walk containing all \(2 \cdot 2^k\) edges, \(E(B_k)\), in the graph \(B_k\).

\[
E(B_k) := \{\langle ax \rightarrow xb \rangle \mid x \in \{0, 1\}^{k-1} \text{ AND } a, b \in \{0, 1\}\}
\]

If \(y \in \{0, 1\}^k\), then \(y = ax\) and \(y = zb\) for unique strings \(x, z \in \{0, 1\}^{k-1}\) and bits \(a, b \in \{0, 1\}\). Then by definition of \(E(B_k)\), there are exactly two edges out of \(y\), one going to \(x0\) and the other to \(x1\), so \(\text{outdeg}(y) = 2\). Likewise, there are exactly two edges into \(y\), one from \(0z\) and the other from \(1z\), so \(\text{indeg}(y) = 2\).

To get from vertex \(b_1b_2\ldots b_k\) to \(c_1c_2\ldots c_k\) with a length \(k\) walk, proceed as follows:
\[
b_1b_2b_3\ldots b_k \rightarrow b_2b_3\ldots b_kc_1 \rightarrow b_3\ldots b_kc_1c_2 \rightarrow \cdots \rightarrow b_kc_1c_2\ldots c_{k-1} \rightarrow c_1c_2\ldots c_k.
\]

Since a walk of length \(k\) exists, a path of length at most \(k\) can be obtained by removing the cycles in the walk.

**Supplemental Problem:**

1The 3-good strings explained here generalize to \(n\)-good strings for \(n \geq 3\). They were studied by the great Dutch mathematician/logician Nicolaas de Bruijn, and are known as \emph{de Bruijn sequences}. de Bruijn died in February, 2012 at the age of 94.

2Problem 9.23 explains why such “Eulerian” paths exist.
Problem 4.
In a round-robin tournament, every two distinct players play against each other just once. For a round-robin tournament with no tied games, a record of who beat whom can be described with a tournament digraph, where the vertices correspond to players and there is an edge \( (x \rightarrow y) \) if \( x \) beat \( y \) in their game.

A ranking is a path that includes all the players. So in a ranking, each player won the game against the next ranked player, but may very well have lost their games against players ranked later—whoever does the ranking may have a lot of room to play favorites.

(a) Give an example of a tournament digraph with more than one ranking.

Solution. Let \( n = 3 \) with edges \( (u \rightarrow v) \), \( (v \rightarrow w) \) and \( (w \rightarrow u) \). Then both \( u, v, w \) and \( v, w, u \) are rankings.

(b) Prove that every finite tournament digraph has a ranking.

**STAFF NOTE:** Hint: Induction on the size of the tournament. Could also rephrase the proof below by considering a maximum length ranking.

Solution. By induction on \( n \) with induction hypothesis

\[ P(n) ::= \text{every tournament digraph with } n \text{ vertices has a ranking.} \]

**base case** \( (n = 1) \): Trivial.

**inductive step:** Let \( G \) be a tournament digraph with \( n + 1 \) vertices. Remove one vertex, \( v \), to obtain the subgraph, \( H \), with the \( n \) remaining vertices. Since removing \( v \) does not change the edges between the remaining vertices, \( H \) is also a tournament digraph. So by induction hypothesis \( H \) has a ranking. Now if the last player in this \( H \)-ranking beat player \( v \), then \( v \) can be added at the end to form a ranking in \( G \). On the other hand, if \( v \) beat the last player in the \( H \)-ranking, then there will (by WOP) be a first player in the \( H \)-ranking that \( v \) beats. Inserting \( v \) just before that first player gives a ranking for \( G \). Since \( G \) was an arbitrary \( n + 1 \) vertex tournament graph, we conclude that \( P(n + 1) \) holds, which completes the proof.