Problem Set 5 Solutions

Due: Wednesday, October 10

Problem 1. [9 points] For each graph in Figure 1, find an Euler tour (if one exists) or an Euler walk (if one exists and there is no Euler tour).

(a) has an Euler walk: c-a-b-c-d-b; it does not have an Euler tour.
(b) has an Euler walk: c-e-a-d-b-f-a-b-c-d-e-f; it does not have an Euler tour.
(c) has an Euler tour: a-c-b-h-i-a-b-i-c-g-h-d-g-f-e-c-d-e-h-a.

Problem 2. [15 points]
In “Die Hard: The Afterlife”, the ghosts of Bruce and Sam have been sent by the evil Simon on another mission to save midtown Manhattan. They have been told that there is a bomb on a street corner that lies in Midtown Manhattan, which Simon defines as extending from 41st Street to 59th Street and from 3rd Avenue to 9th Avenue. Additionally, the code that they need to defuse the bomb is on another street corner. Simon, in a good mood, also tosses them two carrots:

- He will have a helicopter that initially lowers them to the street corner where the bomb is.
• He promises that the code is placed only on a corner of a numbered street and a numbered avenue, so they don’t have to search Broadway.

The map of midtown Manhattan is an example of an $N \times M$ (undirected) grid. In particular, midtown Manhattan is a $19 \times 7$ grid.

Bruce and Sam need to check all $19 \cdot 7 = 133$ street corners for the code. Once they are at a corner, they don’t need any additional time to verify whether the code is there. Once they find the code and return to the bomb, they can disarm it in 2 minutes (even, or especially, as the timer ticks down to 0). Also, they can run one block (in any of the four directions) in exactly 1 minute. They are given 135 minutes total to find the code and disarm the bomb, which means that they need to return to the bomb, code in hand, in 133 minutes.

Sam realizes that the map of NYC is actually a graph, and that they need to use a cool new 6.042 concept: a Hamiltonian cycle is a path that visits each vertex in a graph exactly once and ends at its starting point (so it is a cycle). A graph is Hamiltonian if it has a Hamiltonian cycle.

Hamiltonian graphs are really useful because you can visit each node and return to the starting point by taking only $n$ steps, where $n$ is the number of nodes – if a graph is not Hamiltonian, you would need at least $n + 1$ steps to visit each of the $n$ nodes and return to the starting point.

In general, we don’t know how to efficiently determine whether a general graph is Hamiltonian. However, Sam is very excited because he thinks he can show that Midtown Manhattan is Hamiltonian. If it is, Bruce and Sam can save the day! Will they make it?

(a) [6 pts] Show that they cannot do it – that is, more generally, show that if both $N$ and $M$ are odd, then the $N \times M$ grid is not Hamiltonian. Hint: First show that any $N \times M$ 2-dimensional undirected grid is bipartite.

Solution. Any 2-dimensional undirected grid is bipartite. To show this fact, let us exhibit a coloring of such grid with 2 colors $\{0, 1\}$: indexing the vertices of the grid by their $(x, y)$-coordinates and then coloring vertex $(i, j)$ with color $\text{rem}(i + j, 2)$. The resulting colored grid now has each vertex adjacent to only vertices in the other color. Since any connected graph that can be colored by two distinct colors is bipartite, any $N \times M$ 2-dimensional undirected grid is bipartite.

Suppose the $N \times M$ grid is Hamiltonian. Since $N$ and $M$ are both odd, there is an odd number of vertices in the grid. It follows that the Hamiltonian cycle in this grid is an odd cycle. However, since we have already shown that any 2-dimensional undirected grid is bipartite, odd cycles are not possible. We have reached a contradiction; thus, our supposition that the grid was Hamiltonian is wrong, and we are done.

(b) [9 pts] Suppose Simon defined Midtown in the more standard way as extending from 40th Street to 59th Street and from 3rd Avenue to 9th Avenue (that is suppose Midtown Manhattan was a $20 \times 7$ grid), and gave them another 7 minutes,
1. Show that if either $N$ is even and $M > 1$ or $M$ is even and $N > 1$, then the $N \times M$ grid is Hamiltonian.

**Solution.** Suppose $N$ is even (WLOG). We can write a direct proof or an inductive proof.

**Direct constructive proof:** Assume the grid is laid out on the plane occupying the integer points between $(0,0)$ and $(N-1,M-1)$. We find the Hamiltonian path explicitly by specifying the $k$'th vertex visited for each $k$ from 0 to $NM$. Let $q = \left\lfloor \frac{k}{M-1} \right\rfloor$ and let $r = \text{rem}(k, M-1)$.

On step $k$:
- if $k \leq N(M-1)$ and $q$ is even, then visit vertex $(q, r+1)$.
- if $k \leq N(M-1)$ and $q$ is odd, then visit vertex $(q, M-1-r)$.
- if $k > N(M-1)$, then visit vertex $(0, N-(k-N(M-1)))$.

Checking that it is a Hamiltonian cycle is routine. Figure 2 shows a picture of such a cycle. Thus, we conclude that the $N \times M$ grid is Hamiltonian.

**Inductive proof (also constructive):** We induct on even $N$ values. For the base case $N = 2$, our cycle just has all the exterior edges of the grid. For the inductive step, we assume the existence of a Ham-cycle $H$ on an $N \times M$ grid, and construct a Ham-cycle on the $N+2, M$ grid. Then, consider vertex $v = (n-1,0)$ (this is at one 'corner' of the grid). By the definition of Ham-cycle, $H$ must include $v$, and thus $v$ must 2 distinct edges incident. There are only 2 such edges possible on the grid, and it follows that the edge $((n-1,0),(n-1,1))$ is in $H$. We can remove that edge and add edges $((n-1,0),(n,0)), ((n,0),(n+1,0)), ((n,i),(n,i+1))$ for $1 \leq i < m,$ $((n+1,i),(n+1,i+1))$ for $0 \leq i < m,$ and $((n,m-1),(n+1,m-1))$. See Figure 3. Thus, we conclude that the $N \times M$ grid is Hamiltonian.

2. Explain why your proof breaks down when $N$ and $M$ are odd.

**Solution.** For the direct proof, the odd/even conditions on $q$ require $N$ to be even for the above sequence of vertices to actually form a cycle. For the inductive proof, note that the base case depends on $N$ being 2.

3. Would they survive? Does the outcome depend on where the bomb is placed?

**Solution.** Come on, Bruce and Sam would survive, of course! Their survival doesn’t depend on where the bomb is placed!

**Problem 3. [10 points]** Let the nodes in a tournament be ranked according to their out-degrees, that is, node $u$ is ranked less than or equal to $v$ iff $\text{outdegree}(u) \leq \text{outdegree}(v)$. Prove that the sum of the outdegrees of the $i$ lowest ranked nodes is at least $i(i-1)/2$. 


Figure 2: Hamiltonian cycle described in solution for 2b.

Figure 3: Inductive step for existence of Hamiltonian cycle.
Problem Set 5

Solution. Consider the subgraph on the $i$ lowest ranked nodes. This subgraph represents a tournament as well (see the inductive step in the proof for Theorem 6.2.1 in the textbook). Thus, there is exactly one directed edge between every pair of distinct vertices 1, 2, ..., $i$, and the number of edges in the subgraph is equal to $i(i - 1)/2$. Each of these edges are counted in the sum of the outdegrees of these $i$ nodes in the original graph; it follows that the sum of the outdegrees of the $i$ lowest ranked nodes in the original tournament is at least $i(i - 1)/2$.

Problem 4. [16 points]

(a) [8 pts] Prove that a simple connected graph with $n$ nodes and $n - 1$ edges is a tree.

Solution. To show that a connected graph $G$ with $n$ nodes and $n - 1$ edges is a tree, it is sufficient to show that it is acyclic. Assume to the contrary that there is a cycle in $G$. We can remove the edge in this cycle that preserves connectivity, and continue to remove edges from $G$ until the resulting subgraph no longer contains a cycle; effectively, we are forming a connected acyclic graph $G'$. $G'$ is by definition a tree, but it has fewer than $n - 1$ edges since we removed at least one edge from $G$; thus, we have reached a contradiction and conclude that a simple connected graph with $n$ nodes and $n - 1$ edges is a tree.

(b) [8 pts] Prove by induction that any connected graph has a spanning tree.

Solution. The proof is by induction on the number of edges. Let $P(k)$ be the predicate that if $G$ is connected with $k \geq n - 1$ edges, then $G$ has a spanning tree.

Base Case: $k = n - 1$. Part (a) demonstrates that $G$ is a tree and thus a spanning tree of itself.

Inductive Step: Assume $P(k)$. If $G$ is a connected graph with $k + 1 > n - 1$ edges, it must not be a tree, by Part (a). It follows that $G$ must have a cycle. Removing an edge from that cycle creates a connected acyclic graph $G'$ with $k$ edges, which has a spanning tree over the nodes by our inductive hypothesis. This spanning tree is also a spanning tree over $G$, thus $P(k + 1)$ holds.

By induction, a connected graph $G$ with $k$ edges has a spanning tree, for all $k \geq n - 1$.

Problem 5. [10 points]

Show that the congestion of the $N$-input butterfly is $\sqrt{N}$ if $N$ is an even power of 2.

Solution. First we will show that the congestion is at most $\sqrt{N}$.

Let $v$ be an arbitrary vertex at some level $i$. Let $S_v$ be the set of inputs that can reach vertex $v$. Let $T_v$ be the set of outputs that are reachable from vertex $v$.

Note that:
- For the butterfly network, there is a unique path from each input to each output, so the congestion is the maximum number of messages passing through a vertex for any matching of inputs to outputs.

- If \( v \) is a vertex at level \( i \) of the butterfly network, there is a path from exactly \( 2^i \) input vertices to \( v \) and a path from \( v \) to exactly \( 2^{n-i} \) output vertices.

We thus have \( |S_v| = 2^i \) and \( |T_v| = 2^{n-i} \). The number of inputs in \( S_v \) that are matched with outputs in \( T_v \) is at most \( \min\{2^i, 2^{n-i}\} \). To obtain an upper-bound on the congestion of the network, we need to find the maximum value of \( \min\{2^i, 2^{n-i}\} \), where the maximum is taken over all \( i \). The maximum value is achieved when \( 2^i \) and \( 2^{n-i} \) are as equal as possible. Since \( n \) is even, these two quantities are equal when \( i = n/2 \), hence the maximum congestion is \( 2^{n/2} = N^{1/2} = \sqrt{N} \).

So far, we have concluded that the congestion of \( \sqrt{N} \) can be achieved only at a node at level \( \frac{n}{2} \). Now consider the node at that level whose binary representation is all 0s. Any packet from the input in the form \( \overbrace{0 \ldots 000}^{n/2 \text{ bits}} \) with destination \( \overbrace{000 \ldots 0 z'}^{n/2 \text{ bits}} \), where \( z \) and \( z' \) are any \( n/2 \)-bit numbers, must pass through this node. In the worse case, all packets from input in the form \( \overbrace{0 \ldots 000}^{n/2 \text{ bits}} \) will have destination in the form \( \overbrace{000 \ldots 0 z'}^{n/2 \text{ bits}} \). But there are \( 2^{n/2} = \sqrt{N} \) of such possible packets, giving the node load \( \sqrt{N} \). Therefore, we can conclude that the congestion of \( B_n \) is exactly \( \sqrt{N} \) when \( n \) is even.

Problem 6. [20 points]

In a perfect shuffle, a deck of \( N \) cards is cut exactly in half and then perfectly interlaced. Thus, for \( N \) cards, we would obtain the resulting cards in the following order:

\( 1, (\frac{N}{2} + 1), 2, (\frac{N}{2} + 2), \ldots, (\frac{N}{2} - 1), (N - 1), (\frac{N}{2}), N \).

(a) [10pts] Show that \( m \) perfect shuffles will return a deck of \( N \) cards to its original order provided that \( 2^m \equiv 1 \pmod{(N - 1)} \).

Solution. Let \( N \times m \) switches \( P^1_0, P^1_1, \ldots, P^m_{N-1} \) be available, where \( N = 2^m \) for some positive integer \( m \). In this network of \( N \times m \) switches, there is a one-way link connecting \( P_i \) to \( P_j \), where \( j = 2i \) for \( 0 \leq i \leq N/2 - 1 \) and \( j = 2i + 1 - N \) for \( N/2 \leq i \leq N - 1 \).

Now, let the binary representation of \( i \) be \( b_{m-1}b_{m-2}b_{m-3} \ldots b_0 \), where \( b_k = 0 \) or \( 1 \), for \( 0 \leq k \leq m - 1 \). Then, the binary representation of \( j \) is \( b_{m-2}b_{m-3} \ldots b_0b_{m-1} \), where \( b_k = 0 \) or \( 1 \), for \( 0 \leq k \leq m - 1 \); such that,

\[
i = b_{m-1}2^{m-1} + b_{m-2}2^{m-2} + \ldots + b_12 + b_0,
\]

\[
j = b_{m-2}2^{m-2} + b_{m-3}2^{m-3} + \ldots + b_02 + b_{m-1}.
\]

We see that the binary representation of \( j \) is obtained by cyclically shifting the binary representation of \( i \) one position to the right. For example, when \( m = 3 \), we have
We see in Figure 4, where each of the eight cards is represented by a 3-digit binary sequence, this deck of cards is returned to its original order after 3 perfect shuffles.

Since \(2^m \equiv 1 \pmod{(N-1)}\), we could map \(m\) applications of the perfect-shuffle onto a network of \(m\) columns of \(N\) switches, interconnected by the one-way link described earlier. Given that each binary digit of the input data returns to its original position at the output, \(m\) perfect shuffles will return a deck of \(N\) cards to its original order.

Another approach to this problem: Let us first identify a pattern in how \(N\) cards shift during a perfect shuffle. We have two sets of cards that are produced from the cut; let’s call them sets T and B: T represents the top set/half of cards, and B the bottom set/half. Note that the first card of set B is the original \(0^{th}\)-position card (in this approach we count the stack
of cards from bottom up), and the first card of set T is the formerly \((N/2)^{th}\)-position card. For the sake of simplicity of our future formulation, let us assume that the first card after the first perfect shuffle comes from set B (i.e., the 0th card).

The cards in set B were originally indexed 0 through \((N/2 - 1)\). After the perfect shuffle, card \(i\) in set B will end up in position \(2i\) (all the even positions). The cards in set T were originally indexed \(N/2\) through \((N - 1)\), and each card \(i\) of this set will end up in position \(2(i - N/2) + 1\) (all the odd positions). For easy mapping, we now index both B and T starting with 0: for set B, \((0, 1, 2, 3, \ldots) \rightarrow (0, 2, 4, 6, \ldots)\), and for set T, \((0, 1, 2, 3, \ldots) \rightarrow (1, 3, 5, 7, \ldots)\). Now we need to be able to re-index \(N/2\) through \((N - 1)\) down to 0 through \((N/2 - 1)\), which could be achieved by subtracting off \(N/2\). Hence we replace \(i\) with \((i - N/2)\) to get \(2(i - N/2) + 1\).

We can rewrite \(2(i - N/2) + 1\) as \(2i - N + 1\) and then as \(2i - (N - 1)\). Essentially, all cards, regardless of being in B or T, will be mapped as \(i \rightarrow \text{rem}(2i, N - 1)\): \(N - 1\) comes from the fact that the top card, the \((N - 1)^{th}\)-position card, never moves during the shuffle. Note that the 0th-position card never moves either, as reflected by the mapping \(2 \times 0 = 0\).

Bringing the above mappings and derivations together, the position of card \(i\) after \(m\) shuffles will be \(\text{rem}(i \times 2^m, N - 1)\). This can be rewritten as \(i \times 2^m \equiv r \pmod{N - 1}\), where \(r\) is the resulting position of the card. Great, we are supplied with \(2^m \equiv 1 \pmod{N - 1}\), so we can multiply each side by \(i\) to get \(i \times 2^m \equiv i \pmod{N - 1}\), which means that \(r = i\). Therefore, we conclude that since \(i \times 2^m \equiv i \pmod{N - 1}\), after \(m\) shuffles, card \(i\) will be back to its original position.

(b) [4 pts] Show that 8 perfect shuffles are necessary and sufficient to return a deck of 52 cards to their original order.

**Solution.** Substituting 8 for \(m\) and 52 for \(N\), where \(m\) and \(N\) are related as in Part (a), we find from the congruence below that 8 perfect shuffles are sufficient to return a deck of 52 cards to their original order.

\[
\begin{align*}
2^m &\equiv 1 \pmod{(N - 1)} \\
2^8 &\equiv 1 \pmod{(52 - 1)} \\
256 &\equiv 1 \pmod{51} \\
\frac{256 - 1}{51} &= 5
\end{align*}
\]

We also note that \(2^m \not\equiv 1 \pmod{(N - 1)}\) for \(N = 52\) and \(1 \leq m < 8\). Therefore, 8 perfect shuffles are necessary to return a deck of 52 cards to their original order.

(c) [6 pts] How many perfect shuffles are necessary and sufficient to return a deck of cards to its original order if there are two jokers added to the deck (so that it has 54 cards)?

**Solution.** 52. Note that the top and bottom cards never move. Consider a card \(C\) with
i \in \{1, 2, \ldots, 52\} \text{ cards above. After 52 perfect shuffles:}
\[
\begin{align*}
\# \text{ of cards above } C & \equiv 2^{52} \cdot i \pmod{53} \\
& \equiv 1 \cdot i \pmod{53} \\
& \equiv i \pmod{53}
\end{align*}
\]

The second congruence uses Fermat’s Theorem, which implies that \(2^{52} \equiv 1 \pmod{53}\). If the number of cards above \(C\) is congruent to \(i \in \{1, 2, \ldots, 52\}\), then it must be equal to \(i\). Thus, every card returns to its original position and the deck is restored after 52 perfect shuffles.

Problem 7. [10 points]

For the \textit{Grid, Interrupted} switching network, find the diameter and congestion. Give a reason for your answer.

![Figure 5: Grid, Interrupted](image)

\textbf{Solution.} Diameter is 16. Input \(in_0\) and output \(out_4\) are farthest apart, and the shortest path between the two has a distance of 16.

Congestion is 3. The maximum congestion that will be ever suffered by switches with the most inputs is 3.

Problem 8. [10 points]

Construct a 16-bit de Bruijn sequence.

\textbf{Solution.} We can construct a 16-bit de Bruijn sequence using the labels on the edges of an Euler tour in the de Bruijn graph (see Lecture 9 handout): 0111100101000011.