Mid-Term Experience Math 3310-001

David Browning

October 30, 2016

Sub-Experience 1. The Game. Two players, A and B, alternately select an edge on **the grid graph shown below** and color it red. The loser of the game is the player who is forced to select an edge that creates a red C4 — a red cycle on 4 vertices.

What am I doing?
I will confirm, with proof, that player A (the first player) can always win if she employs a particular strategy for each move.

What do I need in order to do what I'm trying to do?
The graph found both in figure 1 and in the experience.
Figure 2 - 3
The Winning algorithm:
1) Draw a line in the middle of the graph. See figure 2.
2) For every move player 2 makes, mirror it across the x and y axes. See figure 3 for examples of this (Player 1 is the deeper red color, the state of the game is player 2's turn).
3) It becomes painfully apparent after a few rounds of play that player 2 cannot make a move which would cause player 1 to lose as player 1 is mirroring player 2.

How am I going to do it?
Conjecture: If player 1 takes the middle (figure 2), there is no move player 2 can make which cannot be mirrored across both axes.
I'd bet my grade for this problem that I could win any game on this graph (figure 1) if I were granted the first move, but I feel like I come short of an airtight proof as to why. I cannot conceive of a counter example for the above conjecture. If such a move were to exist, the mirror was done incorrectly. Considering the fact that if player 1 is forced to make a losing move while employing the winning strategy, player 2 has already lost, therefore, player 1 will always win.

Sub-Experience 2. The Binary Address Graph. Define the graph Q_n to be the graph whose vertices are all the binary strings of length n and vertices are adjacent if and only if they differ in exactly one entry of the list. The vertices could be thought of as vectors $\vec{v} = (v_1, v_2, \ldots, v_n)$, where $v_i \in \{0, 1\}$; then $\vec{u}\vec{v} \in E(Q_n)$ if and only if $v_i \neq u_i$ for exactly one i.

What am I doing?
I will attempt to prove the following about Q_n:

1. $|V(Q_n)| = 2^n$.

2. Q_n is an n-regular graph; that is, $\deg(\vec{v}) = n$ for each $\vec{v} \in V(Q_n)$.

3. Q_n is *bipartite*; that is, $V(Q_n)$ consists of two sets, say X and Y such that $X \cap Y = \emptyset$ and the only edges of Q_n have one end-vertex in X and the other in Y (so X and Y induce graphs with no edges).

4. Q_n is Hamiltonian for $n \geq 2$.

What do I need in order to do what I'm trying to do?
Figure 4-5
The definition of opposite: for every element e in a verticy, if $e = 0$ in X, $e = 1$ in Y. For example: If 000 is in X, 111 is in Y. If 101 is in X, 010 is in Y.

How am I going to do it?

1. Observe figures 4 and 5. These images represent the cases for $n = 2$ and $n = 3$. Every verticy is of degree $\binom{n}{2}$ since for every item in the binary string there are n things each of which could be either 1 or 0. Every time another element is introduced, two more possibilities are available for every existing verticy, leaving 2^n total vertices.

2. See 1.

3. Let vertex $l \in Q_n$ represent every vertex that has a walk of length 2 from some vertex b. l by definition cannot be adjacent to b. Let b_* be adjacent to some vertex a_*, then $(b_* a_*)$ is the only edge between X and Y where a_* and b_* are merely some particular a and b. Let vertex h represent every vertexthat has a walk of length 2 from a. If you traverse the graph replacing each l with b and each h with a as soon as each possible l and h are visited, sets X and Y will consist of (b, l) (a, h) respectively, and neither X nor Y will have any edges demonstrating that Q_n is bipartite.

4. **Base Case:** n = 2 has a hamiltonian cycle. This seems obvious, but pick any verticy, traverse either left or right, end up at that same verticy.
Assume that for any Q_n graph, there is a hamiltonian cycle.
Let i be some vertex we wish to add to Q_n. i will have an inset and an outset. The outset will have a hamiltonian cycle by definition, and we have assumed that Q_n has a hamiltonian cycle. Therefore for any n, Q_n will have a hamiltonian cycle.

Sub-Experience 3. Spaceship. We have defined (or will define) space stations, and developed (or will develop) notation in class. Please use that notation in your responses to these prompts. To remind: S_w denotes a (specific) w-walled station, $\oint S_w$ denotes its interior, ∂S_w denotes its boundary (its walls and corners). Let's use $V(S_w)$ to denote the corners of S_w and $W(S_w)$ to denote its walls.

Watched Eyes. The function $gg(w)$ is the maximum number of robot eyes (REs) required to protect any w-walled station which can be seen by at least one other RE. Please make a conjecture for the value of $gg(w)$. The points you earn from your response (your conjecture) will be proportional to the type and quantity of justification you give.

Rectangular Stations. Define the function $g_{\perp}(w)$ to be the maximum number of REs required to protect a w-walled station whose interior angles between walls are each either 90° or 270°. Please find the value for $g_{\perp}(w)$ and prove the value you find is correct.

Watched Eyes in Rectangular Stations. Define the function $gg_{\perp}(w)$ to be the maximum number of REs required to protect a w-walled rectangular station such that each RE can be seen by at least one other RE. Please determine the value for $gg_{\perp}(w)$ and prove the value is correct.

What am I doing?

What do I need in order to do what I'm trying to do?

How am I going to do it?

Sub-Experience 4. Some Tournament Problems..

Part One: Curling. Suppose 10 teams compete in a curling competition where each team plays every other team and no ties are allowed. Officials decide that there will be two elimination rounds the first of which will eliminate all teams except for those which beat at least 7 other teams. The second elimination round will eliminate all teams which do not beat a majority of the other teams. The final competition will be among the teams which remain after the second elimination round.

What am I doing?

(A) I will determine with proof the maximum number of teams which may remain after the first elimination round.

(B) I will determine with proof the maximum number of teams which may remain after the second elimination round.

Part One

What do I need in order to do what I'm trying to do?

(A) Figure 7.

How am I going to do it?

(A) There can be at most 5 teams who move on to the next round. There are 10 teams, each of which will play 9 games which means there will be a total of 45 games. 5 teams can all advance solely based upon observation of figure 7 (Note that each vertex is represented by a color which in turn represents a team, and the arrow points to a team they will beat). Notice how all 35 wins can be accounted for by calculating 2 losses per team who advance in the tournament, and 5 losses per team who does not advance (leaving the expected 35 wins). If we were to hypothetically augment this to 6 teams who move on, and by that same logic is impossible: 6 advancing teams times 2 losses per team = 12 + 4 teams not advancing times 6 losses per advancing team = 32 available losses where we would need 6 times 7 or 42 losses available in order to advance 6 teams.
Math form:
$5 \times 7 = 35$
$5 \times 5 = 25$
$5 \times 2 = 10$
$10 + 25 = 35$
$35 =$ The number we need!

$6 \times 7 = 42$
$4 \times 6 = 24$
$6 \times 2 = 12$
$24 + 12 = 32$
$32 =$ insufficient!

(B) This is the same problem with different numbers.
Total Games: $(\frac{5\times4}{2} = 10)$.
Wins needed to advance: 3
If three teams advance, $3 \times 3 = 9$. We need at least 9 losses from the reamining pool of games. $3 \times 1 = 3 + 2 \times 4 = 11$. We have 11, and $11 > 9$, So we have at least a lower bound of 3 teams advancing.
Let's try four.
$4 \times 3 = 12$. We need 12 losses from the remaining pool of games. $4 \times 1 = 4 + 4 \times 1 = 8$, $8 < 12$. 4 teams cannot advance; Three teams is therefore the largest number of teams that could advance.

Part Two: Strong Tournaments. Recall that a directed D graph is **strong** if between any pair of vertices x and y, there is an x, y-path, and a y, x-path.

What am I doing?

(A) I will prove that in any strong tournament T on at least 4 vertices, there exist two distinct vertices x and y such that $T - x$ and $T - y$ are strong. Prove or disprove whether the conclusion "$T - x - y$ is strong" holds as well.

(B-1) Let T be a tournament on $n \geq 3$ vertices and let s be a vertex of T. Please prove the following statement: *T is strong if and only if for every vertex $t \in V(T) \setminus \{s\}$ there is an s, t-path and a t, s-path.* Is this statement true if T is a digraph (not necessarily a tournament).

(B-2) Suppose you have an algorithm $\mathcal{A}$ that determines whether there is a path from x to y, where x and y are vertices in a digraph. Suppose $\mathcal{A}$, when given a pair of vertices and a digraph D, performs this calculation with M operations in the worst case. How many operations, in the worst case, are needed to use $\mathcal{A}$ to determine whether D is strong using the standard definition of 'strong'? How many operation, in the worst case, are needed if the alternative definition of 'strong' proved in B-1?

What do I need in order to do what I'm trying to do?

Figure 8.

How am I going to do it?

(A) It seems obvious that figure 8 is a worst case scenario with four vertices with regard to keeping a tournament strong while removing at least one vertex. Notice that if we let x be the bottom left vertex in figure 8 and y be the bottom right, The statement in A is true. This is the case because removing a single vertex renders the tournament no longer strong (top-left). I argue that 4 vertices (and specifically figure 8) is also a worst case scenario for any number of vertices greater than 4 as adding more vertices can only render keeping the tournament strong easier or as easy as four. The only counter example would be a strong tournament with greater than 4 vertices where removing 0 vertices renders the tournament no longer strong. In addition, adding vertices can only either take one of the k_3's and turn it into a $k_{somethingGreaterThan3}$ or do nothing to the k_3's making the added vertices irrelevent to the problem.

(B-1)

(B-2)

What do I need in order to do what I'm trying to do?

How am I going to do it?

Sub-Experience 5. An Optimization Problem.

Let H denote an arbitrary graph. Recall that the distance between vertices in H is the length of the shortest path that has those vertices as its endpoints. Denote by $d_{\max}(H)$ the maximum distance among all pairs of vertices of H. Recall that $\Delta(H)$ denotes the maximum degree among all vertices of H.

Define the function $N(d, k)$ to be the maximum number of vertices among all graphs H with $d_{\max}(H) = d$ and $\Delta(H) = k$.

One. *Determine* $N(n, 2)$.

Two. *Determine* $N(2, 3)$.

Three. *Verify that the graph* G *in Figure* **??** *has* $d_{\max}(G) = 2$.

Four. *Determine* $N(2, 4)$.

What am I doing?

What do I need in order to do what I'm trying to do?

How am I going to do it?

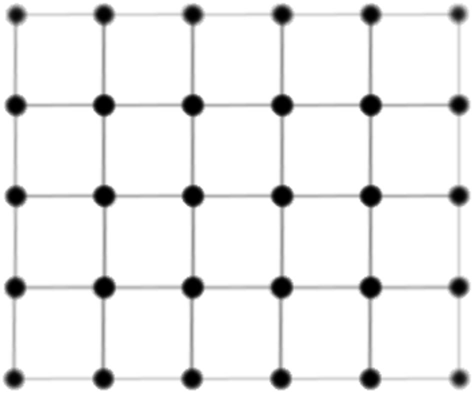

Figure 1: The Graph

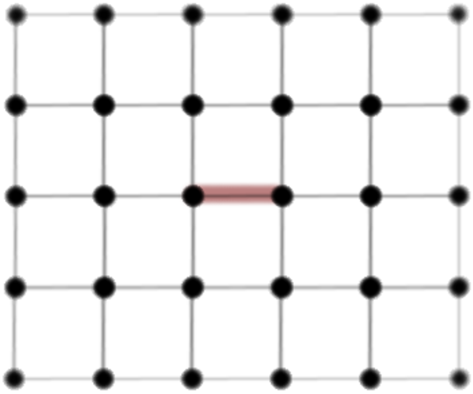

Figure 2: The First Move

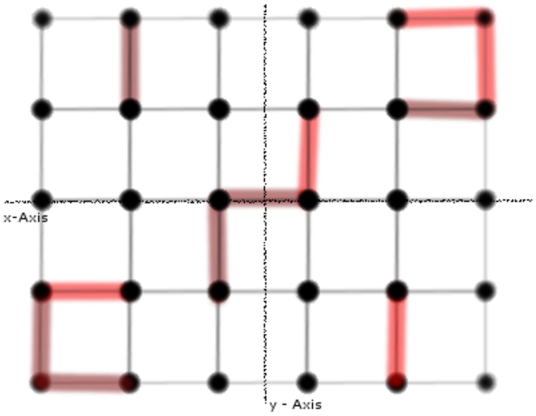

Figure 3: Mirrored moves

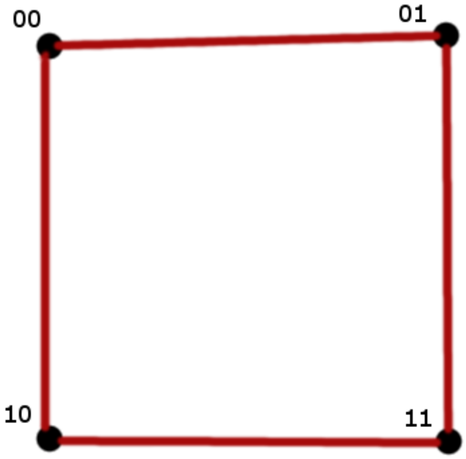

Figure 4: Base Case Binary Strings

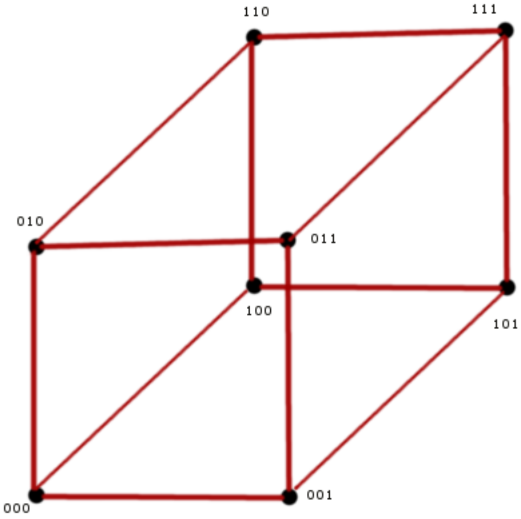

Figure 5: Base Case + 1

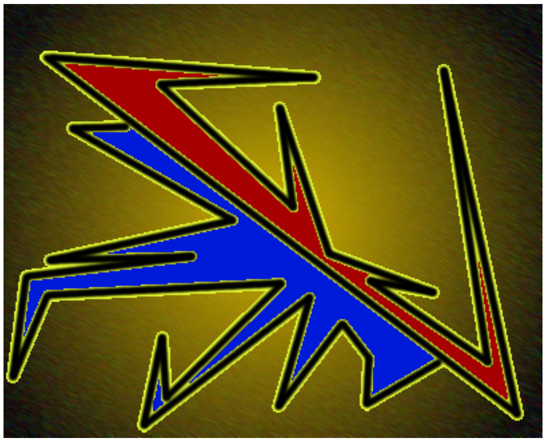

Figure 6: Space Station

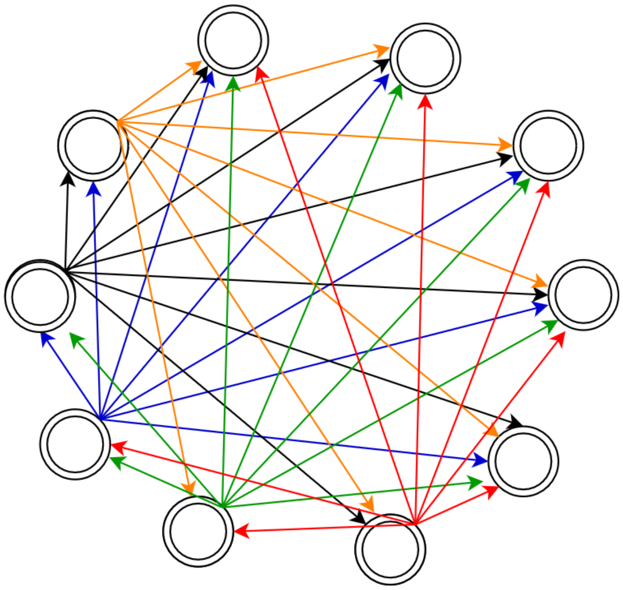

Figure 7: Five Winners

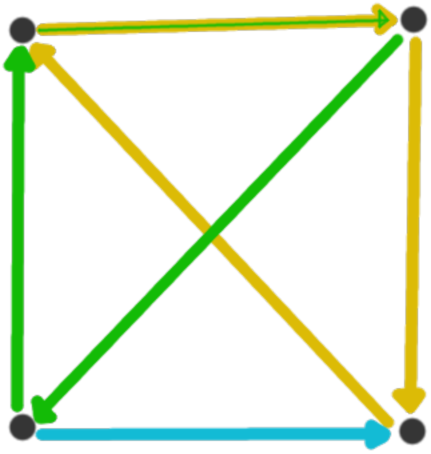

Figure 8: Directed Graph n = 4