

TOURNAMENT TRAINING VIDEO SERIES TRANSCRIPT

INTRODUCTION

Hello, and welcome to the Tournament Training video series, presented by the National Hispanic Institute at Tip of Texas. In this video series, we will be explaining the methods and approaches to some of the most important tasks that must be completed in order to carry out a Great Debate tournament.

In several instances, you will be asked to complete exercises and apply the information that has been presented—we encourage you to always attempt to answer these exercises, as they serve to reinforce your understanding of the material and can also help to identify gaps in your comprehension.

Our goal is for you to be confident in your ability to execute your responsibilities as a member of the Tournament Staff, so if you have any questions about the content of these videos, please do not hesitate to visit our website, tipoftexasnhi.org, and submit them through our contact page.

That concludes this introductory video. In a few moments, you will be directed to the first instructive video.

CODES

Hello, and welcome to this installment of the Tournament Training video series, presented by the National Hispanic Institute at Tip of Texas. In this video, we will be discussing codes—what they are, how they are created, and how they are assigned.

In a tournament, instead of referring to competitors and judges by their names or the names of their teams, we use codes: combinations of letters and numbers that make it easier for us to identify and organize them.

There are three types of codes: the first type are Team Codes. A team code consists of a single letter—the first letter of the official team name. If two or more teams' names begin with the same letter, then an alternative team code will be assigned at the discretion of the Tournament Director. So for example, the team code for Tip of Texas would be "T," the team code for Houston would be "H," and the team code for Corpus Christi would be "C."

The second type are Competitive Entity Codes. First of all, the term "competitive entity" refers to a person or pair of people that compete in a category at the Great Debate—the term is used to avoid the confusion that may result from referring to both sets of competitors and the teams they belong to as "teams." A competitive entity code consists of a letter—the team code of the team they belong to—and a number—their place in their team's roster for their category.

So for example, the first Competitive Entity in the Laredo Oratory Roster would be given the Competitive Entity code L1 and the fourth Competitive Entity in the Dallas Mock Trial Roster would be given the Competitive Entity code D4. Note that the category is not taken into account when determining a Competitive Entity code.

And the third and final type of code are JFL Codes. A JFL code consists of a letter—the team code of the team the JFL belongs to—and a number—their place in their team’s JFL roster. So for example, the second person in the JFL roster from Austin would be given the JFL code AA2 and the sixth person in the JFL roster from El Paso would be given the JFL code EE6.

Please take a few moments to complete this exercise—feel free to pause the video if you need more time. Time’s Up. The team code for Fort Worth would be F, the first letter of their team name, the Competitive Entity Code for the third team in the Oratory roster from Panama would be P3, the first letter of their team name and their place in their team’s roster for their category, and the JFL code for the Second Person in the JFL roster from Baytown would be BB2, the first letter of their team name doubled followed by their place in the roster.

And now for the easier part, assigning the codes. All codes are assigned after the team’s Official Roster has been submitted. So let’s say we received the roster for Tip of Texas and wanted to start with Oratory. We would first import the roster into our spreadsheet application of choice, then we would Insert an empty column to the left of the Competitive Entities like so, and then fill those empty cells with the codes of the Competitive Entity in the cell to the right of them. And we would repeat this process for the JFLs. First import into our spreadsheet application, insert an empty column to the left of the competitive entities, then fill those empty cells with the codes of the Competitive Entities to the right of them.

That concludes this installment of the Tournament Training video series. In a few moments, you will be directed to the next video.

MATRICES

Hello, and welcome to this installment of the Tournament Training video series, presented by the National Hispanic Institute at Tip of Texas. In this video, we will be discussing matrices—what they are and how we use them to create matchups.

A matrix is a table composed of a series of cells in which competitive entities and round numbers are placed in order to create matchups for a particular category. There are two types of matrices: the first type we will look at is that which involves an even number of competitive entities. Here are the steps to creating matchups using a matrix.

The first step is to place the Competitive Entities on the axes. The procedure for doing this is the following: first we list the competitive entities in numerical-alphabetical order, now what that means in practice is that we would take all of the competitive entities in a category and first list all those whose code ends in 1 in alphabetical order, then all those whose code ends in 2 in alphabetical order, etcetera. So for example, let's say that this was the category roster we were given. As you can see, the total number of competitive entities is even—12—and each has already been assigned their code. In order to place these on the axes, we would first list them in numerical-alphabetical order, which would look like this. The competitive entities are ordered in numerical order, with the codes with the number one followed by those with the number two and so on, and they are also in alphabetical order: in each numerical set, the codes with the letter A are followed by those with the letter B, and so on.

Next, we would place half on the horizontal axis and half on the vertical axis. As we mentioned before, there are a total of 12 competitive entities, so we would place the first six on the horizontal axis and the second six on the vertical axis. Before we do that, let's take a look at the matrix we've prepared. Taking into account the total of 12 competitive entities that are competing in this category, we've left 6 cells where the first half will be placed and six cells where the second half will be placed. Now we are ready to place them on the axes according to our procedure, and this is what that looks like. As you can see, we placed the first six competitive entities—A1, B1, C1, A2, B2, and C2 on the horizontal axis and the second six—A3, B3, C3, A4, B4, and C4 on the vertical axis, all in numerical-alphabetical order.

Please complete this exercise—feel free to pause the video if you need more time. Time's Up. In numerical-alphabetical order, the sequence of Competitive Entities would be A1, B1, C1, D1, A2, B2, C2, D2—we arrive at this by listing all the codes that end in one in alphabetical order and then all the codes that end in two in alphabetical order.

And now onto step 2: identify the impossible. In any matrix, each cell that does not contain a Competitive Entity represents a matchup—and the matchups that each cell represents are determined by the competitive entity above and to the left of that cell. For example, the cell highlighted in yellow represents a matchup the matchup B2 versus C3, because the competitive entity immediately above it is B2 and the competitive entity to the left of it is C3. In a sense, this cell is where those two competitive entities intersect. In this matrix, that cell represents B2 versus C3, and B2 versus C3 is one possible matchup—a competitive entity from Team C competes against a competitive entity from team B. But not all matchups in this matrix are possible.

The rules governing the arrangement of Round-Robin rounds are the following:

1. No competitive entity can compete against another competitive entity more than once.

2. No competitive entity can compete against another competitive entity that belongs to the same team.

Luckily, the structure of a matrix prohibits repetitive rounds, so we need not worry about that and can thus focus on making sure the second rule is not violated.

Let's go back to our matrix; the most efficient way to identify all impossible matchups is to begin with the impossible matchups that are closest to the competitive entities and move downward and to the right until we reach a matrix border—as you will see, because the competitive entities are listed in numerical-alphabetical order, impossible rounds tend to appear in diagonal patterns.

Let's begin with this impossible matchup, A1 versus A3—because the two competitive entities share a team code, meaning they are from the same team, this matchup is not possible, since it is in violation of Round Robin Rule #2. We mark impossible rounds by placing an X in that cell like so. As we said before, the most efficient way to identify all impossible matchups is to move downward and to the right until we reach a matrix border, so we will do exactly that: let's look at the cell that is downward and to the right from this one, B1 versus B3—this matchup is impossible for the same reason, so we will place an x there. We once again move downward and to the right. C1 versus C3? Not possible—x. Downward and to the right. A2 versus A4? Not possible—x. Downward and to the right. B2 versus B4? Not possible—x. Downward and to the right. C2 versus C4? Not possible—x. And at this point, we can no longer go downward and to the right since we have reached the matrix border. So we now look for another impossible round that is close to the competitive entities.

Let's go with this one, A2 versus A3. Not possible—x. Downward and to the right. B2 versus B3. Not possible—x. Downward and to the right. C2 versus C3. Not possible—X. And we have once again reached a matrix border, so we look for another impossible round that is close to the competitive entities. A4 versus A1. Not possible—x. Downward and to the right. B1 versus B4. Not possible—x. Downward and to the right. C1 versus C4. Not possible—x. And we have reached another matrix border, except this time there are no more rounds that are not possible—we have identified all of the impossible.

Before moving on to step 3, please complete the following exercise—feel free to pause the video if you need more time. Time's Up. Here is the matrix with all impossible matchups identified. At this time, we will move on to step 3: arranging rounds.

The objective of arranging rounds is simple: give each competitive entity one opponent for round 1, one opponent for round 2, one opponent for round 3, and one opponent for round 4. In our matrix, what this means is that we need to find an uninterrupted diagonal or pair of diagonals that

does not overlap and runs parallel to the possible diagonals. By diagonals, we mean patterns similar to those that were created by the impossible rounds, except when arranging rounds, what we want to create is a pattern of possible rounds. Let's return to our matrix.

In any matrix, the easiest uninterrupted and non-overlapping diagonal that can be made is the central line—this one, but in this case, it is occupied by impossible rounds, so it is not a viable diagonal. But we can try the diagonal adjacent to it—this one. All of the cells that this diagonal intersects are possible rounds, it is uninterrupted, and runs parallel to the possible diagonal—great, so we have our first round, right? Well, let's take another look. We said earlier that our objective was to give each competitive entity exactly one opponent for each of the four rounds, but who would A1 and C4 be going against? No one—they have not been assigned a round. And this leads us to an important concept: any diagonal that is not central must be complemented by another diagonal. By central, we mean that it stretches from corner to corner, like this one. Returning now to the viable diagonal we had previously found, we need to complement this one with another one that makes it so that A1 and C4 have a round. How about this diagonal? With this diagonal, A1 and C4 now have an opponent—A1 would compete against B4, a possible round, and B1 would compete against C4, another possible round. Let's think about our objective once again: to give each competitive entity exactly one opponent for each of the four rounds. Take a look at competitive entity B1. If we accept the highlighted diagonals as a round, that would mean that it would have not one but two opponents for that round—A3 and C4. In other words, the pair of diagonals overlaps and creates an impossible situation—a competitive entity cannot compete against more than one competitive entity at a time.

What about this one? Take a look at this pair of diagonals, is it viable for a round? The answer is yes, it is. The diagonals are uninterrupted, they do not overlap, and they run parallel to the impossible diagonals—we have our first round, accordingly, we will place the number “1” in the cells that these diagonals traverse, like so.

And we repeat this process for round 2. What about this pair of diagonals, is it viable? No, it is not, it does not run parallel to the impossible diagonals and it traverses not one but two impossible rounds. What about this pair of diagonals, is it viable? No, it is not, the diagonals overlap, and as a result, both B1, C1, and A2 would each have two opponents. And this pair of diagonals, is it viable? Yes, it is—the diagonals are uninterrupted, they do not overlap, and they run parallel to the impossible diagonals—we have our second round, accordingly, we will place the number “2” in the cells that these diagonals traverse, like so.

And we repeat this process two more times. Round 3: are these diagonals viable? Yes. Round 4: are these diagonals viable? Well they better be, because there are no others left, but don't worry, you will almost never run out of options—we did in this case only because it is a relatively small matrix.

Now that we have arranged all four rounds, we can now move on to step 3: double checking. We need to double-check that we have met our objective—that each competitive entity has been assigned one opponent for round one, one for round 2, one for round 3, and one for round 4. On our matrix, what that means is that each column should have one number one, one number two, one number three, and one number four in it. So let's check.

Does A3 have exactly one round 1, one round 2, one round 3, and one round 4? It does. What about B3? It does as well. And we continue to check until we reach the last competitive entity on the vertical axis, at which point we do the same for the competitive entities on the horizontal axis. And once we see that each has exactly one round 1, one round 2, one round 3, and one round 4, we color code each of the rounds—the reason for this will become apparent in the next video.

Please take a few moments to complete the following exercise—you may pause the video now and unpaue when you have completed the exercise. Four diagonals are possible, and here they are highlighted in different colors—note that it does not matter which is round 1, 2, 3, or 4.

Now that we have learned the procedure for creating the first type of matrix, we will now move on to the procedure for creating the second type of matrix—one that has an uneven number of competitive entities. Fortunately, we follow a nearly identical procedure, with only one slight change.

Let's say this was our category roster. If you count up all the competitive entities, you'll see that there is an odd number of them—15 to be exact, but we will follow the same procedure we used for the even matrix.

Step 1: Place on Axes. First we list them in numerical-alphabetical order, like so, and secondly, we place half on the vertical axis and half on the horizontal axis. But wait—if there is an uneven number of competitive entities, how can we divide them in two? And also, if they are uneven, doesn't that mean that it is impossible for each and every one of them to have an opponent? The answer is yes, in an uneven matrix, there will be exactly one competitive entity every round that does not have an opponent. The protocol in this situation is the following: we go to our list of competitive entities, and we add a "BYE" round. In the matrix, we will treat this BYE round as if it was another competitive entity., the only difference is that when a competitive entity is matched-up with a "BYE" entity, what that means is that for that round, they will have a BYE round, meaning that they will not be competing, and instead, their score will be the be calculated by averaging their other three scores.

With the addition of this BYE round, we can now treat this uneven matrix as if it were an even one. We place these eight entities on the horizontal axis, and these eight entities on the vertical axis.

Step 2: Identify the impossible. As we did before, we begin with the impossible rounds near the edges and work our way downward and to the right.

Step 3: Arrange the rounds. We begin with the easiest diagonal, the central one, and then proceed to look for other uninterrupted diagonals that do not overlap and run parallel to the impossible diagonals.

And after completing step four and double-checking our matrix, we color code each round.

That concludes this installment of the Tournament Training video series. In a few moments, you will be directed to the next video.

ASSIGNING JUDGES

Hello, and welcome to this installment of the Tournament Training video series, presented by the National Hispanic Institute at Tip of Texas. In this video, we will be discussing the process of assigning judges.

A quick note: we use the term “judge” to refer to a person who oversees, scores, and time-keeps one or more rounds at a Great Debate tournament—we do not use the term “JFL” because, although almost all judges are JFLs, not all JFLs are judges; at most tournaments, the on-site director will have a few JFLs help them with logistical matters and the tournament director will recruit a few to serve as bracket coordinators. All this to say: when you’re assigning rooms and rounds to judges, make sure that they are in fact available to act in that capacity. And now onto the process of assigning judges.

The first step is to assign each judge a room. This is the easiest step in the process. First, we create a master table that lists every person from every team who will act as a judge along with their JFL codes. Next, we add an empty column to the right, and fill those cells with the rooms where they will be judging.

If it is necessary, you can always change the rooms at a later time, but it is best to do so as infrequently as possible—this is because in most instances, judges are adjudicating rounds back-to-back, and it is more time-efficient for a judge to not have to move to another room or another building in between those rounds.

The second step is to transcribe rounds. First we create a table that is formatted like this, with one four columns—one for each round—and two cells in each one of the rows; then we place our matrix somewhere nearby so that we can easily reference it—we'll be using the uneven matrix from the video on matrices. And now we are ready to begin to copying the matchups onto the table.

This is what that process looks like: we begin with the first round, and we will begin with this matchup—this matchup is A1 versus C3, and so we copy that into the row labeled “R-1” for round one, like so. We do the same thing for the next matchup, A4 versus B1, repeating the procedure. And we work our way across the matrix, making sure we account for every single matchup in the first round, and once we're done with round 1, we move on to round 2, then round three, and finally round 4, until all rounds are transcribed and placed in the table.

Step 3: Assign rounds. As you'll soon see, the process of assigning judges to rounds is really the process of assigning rounds to judges. And we do that according to two rules for Round Robin Judge Assignment: (1) no person can judge a competitive entity that belongs to their team—meaning that they cannot share a team code—and (2) no person can judge the same competitive entity more than once.

With this in mind, we can begin to assign rounds to judges. Our set up is the following: we have the table with the rounds transcribed on one side and a nearly identical table on the other, except this one has our judges' codes listed in a column on the left.

The process looks like this. We begin with round one, with the first matchup in the list: A1 versus C3. We need to assign this matchup to a judge, meaning that we will need to place it next to their code under the column labeled “R-1” for round 1. Could we assign it to AA2? No, because AA2 and A1 share a team code, meaning they are from the same team, and a judge cannot adjudicate a round that involves a competitive entity belonging to the same team. What about CC2, could we assign this matchup to that judge? The answer is no and for the same reason—C3 and CC2 are from the same team. What about BB1? Yes, we could assign this round to BB1—and so what we do is cut and paste the matchup into the round 1 column in the row to the right of BB1, like so. And we repeat this process with the rest of the rounds—assignments for round 1 are the easiest, since we do not need to worry about someone judging the same team more than once—we need only make sure the competitive entities and their judges are not from the same team.

And we continue the process with the matchups from round 2. B1 and C3—could we assign that matchup to BB2? No, because B1 and BB2 share a team code. What about BB1? No, because not only do BB1 and B1 share a team code, BB1 has already judged C3. And what about AA1? Could we assign B1 and C3 to them? Yes—they do not share a team code and AA1 has judged

neither B1 or C3. And again, we repeat this process until we have assigned all rounds for rounds two, three, and four.

Step 4, double-check, is particularly important for Judge Assignments. On the day of Round-Robin rounds, the majority of complications arise from improperly assigned judges. The way we double-check judges assignments is by looking at our completed table and checking row-by-row that each judge does not adjudicate a round involving a competitive entity from their own region and that they do not judge any competitive entity more than once.

Let's take a look at AA1—are there any competitive entities in this row that begin with the letter A, meaning that they are from the same team as AA1? No. Does C1 appear more than once? No. What about B4, does it appear more than once? No. And B1? No. And C3? No. And B2? No. And C5? No. And we repeat this for each and every row.

The last step is to fill in the matrix. This process is straightforward—all we do is place our judges in the cell that represents the matchups they will be adjudicating. Our set-up is the following: we have our completed table on one side and our matrix on the other. We start with the first round. AA1 is set to judge C1 and B4, so we look for where this matchup is on our matrix. It's right here—and all we do is replace the number one with the code of the judge who will be adjudicating that matchup, like so. And we repeat this process with all matchups for all rounds until we end up with something that looks like this. This is the reason why we color coded each round earlier: in the absence of the numbers, the colors tell us which matchups are from round 1, which are from round 2, which are from round 3, and which are from round 4.

This concludes this installment of the Tournament Training video series. In a few moments, you will be directed to the next video.

FROM SPREADSHEETS TO BRACKETS

Hello, and welcome to this installment of the Tournament Training video series, presented by the National Hispanic Institute at Tip of Texas. In this video, we will be discussing how to translate the matrices and judge assignments we've made into brackets for participants and staff members to use.

First, we create a template for our bracket—there are several styles that are acceptable, but this is the one we will be using today—it was made in a few minutes in Google Sheets, but any spreadsheet application or word-processing application should do.

In this bracket, each one of these fork-shaped figures represents a round—this is a round, this is a round, and this is a round.

After we have created our template where we will write the competitive entities and judges, the first step is to copy the codes onto the brackets, and we do that following this format: on the top level of the fork we place one competitive entity, on the bottom level the other competitive entity, on the handle we place the judge, and below that we place the room where the round will take place—again, we are first copying the codes, not the names.

For this example, we'll be using the judge assignments we created in the “assigning judges” video, which was based on the uneven matrix from the video on matrices. We would start with the first matchup of the first round—this one: C1 competes against B4 and AA1 judges that round. So we would place one of those competitive entities, C1, on the top level, the other, B4, on the bottom level, the judge, AA1, on the handle, and we would reference our master roster to see which room AA1 was assigned to—room M114—and we would place that right underneath. And we would follow this same process to copy the rest of the matchups and judge assignments onto the bracket like so.

After copying the competitive entity and judge codes, we are now ready to replace these codes with the names of the people they denote. Before we do this, it is important that we keep a copy of the bracket that has the codes on it, preferably one that can be easily and quickly edited. The process for replacing codes with names is simple: we place our master rosters nearby for reference and we substitute the competitive entities' codes for the names linked to them. And what we will end up with is something that looks like this.

As always, the last step is to double-check. What we double check is (1) that there was no mistake committed in copying the codes from the table with the judges assignments to the bracket and (2) that there was no erroneous substitution of codes for names—we do this using the coded bracket we saved earlier.

This concludes this installment of the Tournament Training video series. In a few moments, you'll be directed to the next video.