*Break into POGIL teams of 4 and assign each team member one of the following roles.*

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Role** | **Responsibility** |
| Cole Swierczek | Facilitator | Manages interaction with the Parity Magic widget to help test the team’s algorithms. |
| Grayson Kurth | Spokesperson | Reports the team’s results to the teacher and other teams. |
| JP Duffy | Quality Control | Records all answers & questions, and provides the team’s reflection to team and instructor. |
| Zachary Lineman | Process Analyst | Considers how the team could work and learn more effectively. |

Error Correction: Critical Thinking Questions

Each team will receive a deck of cards. For a regular card deck you can use face-up/face-down to represent 0/1. A satisfactory outcome for this activity is that the team can successfully demonstrate the trick to the class. That means, someone will lay out a 5x5 array of cards randomly. Then a member of the team will layout the 6th row and column and will successfully identify the flipped card when some from the class secretly flips a single card.

1.) In the video, are the 6th row and 6th column being laid out in a truly random way or is some kind of rule or algorithm being used? If so, what's the rule?

There is an algorithm used to determine what the value of the 6th row and column are. The algorithm checks if the amount of green androids is even if it is not even it makes the certain row or column green to make it an even number or black if the 1st through a 5th row or are already even.

2.) HINT: Count the number of face cards in each row and column. What pattern or rule do you see if you do that?

The number of face cards in each row or column is always even.

3.) Practice: Everyone on the team should practice the "trick" using the widgets or the deck of cards.

**DONE!**

4.) (**Portfolio**) What is the "trick"? Of course, it's not really a trick. It's an algorithm. So, describe an algorithm in pseudocode that solves the problem of identifying the flipped card.

Repeat through all columns: {

IF (AmountOf1sinColumn = odd) {

Then IF (AmountOf1sinRow = odd) {

FlippedCard == True

}

}

}

5.) (**Portfolio**) The card "trick" shows that it is always possible to identify the card that was flipped as long as only one card was flipped. Would it be possible always to determine if an error occurred if two cards were flipped? Experiment with the cards or widgets to help answer this question.

It is not possible to detect if two bits were flipped using this algorithm because if you flip two bits in the same column or row it will not be able to determine what to change and if you flip two different bits in different columns and rows it will think four bits have been flipped because there will be four columns and rows (2 each) that will change in order to detect the error and these will intersect with each other, creating the illusion there is four.

6.) In this case, the 25 original cards (bits) are data and the 11 additional bits are for error detection, meaning that 25/36 = 69% of the bits are data and 31% are redundant bits used for error detection. Suppose the original array was 3x3. How many error detection bits would you need in that case and what percentage of the total bits would now be data bits?

7 error detection bits. 9/16= 56.25%