CSCI 2824: Discrete Structures (Spring 2011).
Instructor: Sriram Sankaranarayanan
Notes for Lecture # 1

# 1 Introduction

Introduction: instructor and learn about student backgrounds.
Course details:

**Web Page:** `https://www.cs.colorado.edu/~srirams/classes/doku.php/discrete_structures_spring_2011`.

**Moodle Signup:** Talk about signing up on moodle. Enrollment code: `csci2824-s11`.

**Book:** Warn about student solution manuals. Other books they may want to use?

**Assignments:** Weekly assignments (discuss late policy, collaboration policy), programming assignments, reading assignments.

**Exams:** Three midterm (worst score dropped), Final (compulsory, $\geq 25\%$ required to pass course).

# 2 Parity and the Prisoners Hat Puzzle

Imagine there are 10 prisoners and an evil warden. Just for sport, the warden lines up the prisoners, one behind the other, each prisoner facing the back of the prisoner in front. He then places a hat on each of them. The hat can be colored red or green.

Each prisoner can see the colors of the hats in front of him, but **not** the color of his own hat or those behind him. After the hats have been placed on each prisoner, the warden asks each prisoner to name the color of his hat, starting from the last prisoner (who can see all the other hats but his own). Any prisoner who cannot name the color of his hat is executed.

But wait, before you feel sorry for them, you should know that these are really smart prisoners we are dealing with here!

## 2.1 Version 1

Suppose, just before this cruel game begins, they see a pile of 5 red hats and 5 green hats (which eventually end up on their heads). How can they ensure that no one is killed?

## 2.2 Version 2

They have no information on how many red/green hats there are. How can they ensure that at least 9 of them survive?

## 2.3 Version 3

Ignore this if infinite sets are not your cup of tea (yet)!!

If there are infinitely many prisoners (say countable) then is there a scheme that ensures that only finitely many prisoners die?

# 3 Parity Bits

Turns out this puzzle is loosely related to a field of CS called *Coding Theory.* Using coding theory, we make sure that information can be stored and transmitted, so that even if they get corrupted, we can detect the corruption and/or recover the original information.

Computers transmit bit sequences over networks. It is possible that while a bit sequence is transmitted over a cable or a wireless channel, an error can flip a single bit or many bits. Often, depending on the situation, the single bit being flipped is much more probable than two or three bits being corrupted.

So if computer A intended to transmit the string "1000101" it is possible that the third bit from the left gets flipped to yield "1010101". An interesting question is to use a *code* to detect if there is an error. A commonly used code is called a *parity bit*.

The idea is simple: we tack on an extra bit at the end that specifies if the number of 1s is odd. In other words, the extra bit is a "1" if the number of bits in the original string is odd and a "0" if it is even.

Eg.,

- Original 7-bit sequence: "1000101"
- Bit sequence with parity bit: "10101011"

The extra bit here is a parity bit that says that the number of 1s in the first 7 bits is odd.

Using parity, we can detect if a one-bit flipping error has occurred during transmission. How? Suppose, if the third bit flipped due to a transmission error, we get "10101011".

Looking at the parity bit, we see that it is a "1" even though the bit sequence has an even number of "1"s. Therefore, we conclude that an error has happened.

Questions to consider:

1. What happens if two bits get flipped? Will the parity scheme still work?
2. What happens if the parity bit itself is flipped?
3. Can you provide a scheme that can detect any single bit flip during transmission including the parity bit(s)?

# 4 Solution to Prisoner's hat puzzle

We now discuss solutions to both versions.

## 4.1 Version 1

Version 1 is simpler. Recall that there are 5 red hats and 5 green hats. We have to reason by putting ourselves in the place of the prisoner and reasoning it out.

**Last prisoner:** This prisoner sees the colors of the 9 hats in front of him. He knows there are 5 red/5 green. So he can accurately guess what his hat color should be.

**Any other prisoner:** This prisoner has heard the hat colors of the prisoners behind them and can see the hats of the prisoners ahead of them. Therefore, knowing that there are 5 red hats and 5 green hats, the prisoner can say the color of their hat accurately.

## 4.2 Version 2

Here is where parity comes in handy. The prisoners meet up before they start and hatch the following scheme:

**Last prisoner:** This prisoner is unfortunately going to have to sacrifice himself for his friends. He looks ahead and says "RED!" if there are an odd number of red hats in front of him and "GREEN" if the number of red hats is even. In other words, he computes the *hat parity*. This prisoner has roughly 50% shot of being right.

**Any other prisoner:** They know the colors of the hats in front of them. They know the colors of all the hats behind them (except for last prisoner) and the "parity" information from the last prisoner. Convince yourself that this is enough information to work out the color of their own hats.