CS 740: Assignment 2 - Null ho na ho

- Announced: Aug 27, Posted: Aug 28, Due: Sep 05 11:50 AM
- Please write (only if true) in your handwriting the honor code. If you used any source (person or thing) explicitly state it.
- **This is a team assignment.** Please see the piazza post on working in teams on how to ascribe percentages to group effort
- You should not use any code from the internet. That violates the honor code.

Overview

There are three parts in this assignment which are based on concepts already covered in class. In the first part, we return to traditional Lights Out to make sure you understand the intuition behind the subspaces.

The next two parts describe two related programming problems. You are required to write python programs to solve the problems described in these parts. Note that a skeleton python program has been provided to you that you could use to solve the two problems. You cannot use any other code that you are not an author of, nor should you actively seek code to modify. The only reason to use the Internet is to look up documentation on python.

It is advisable to attempt the parts in the order given.

Lights Out: Part 1

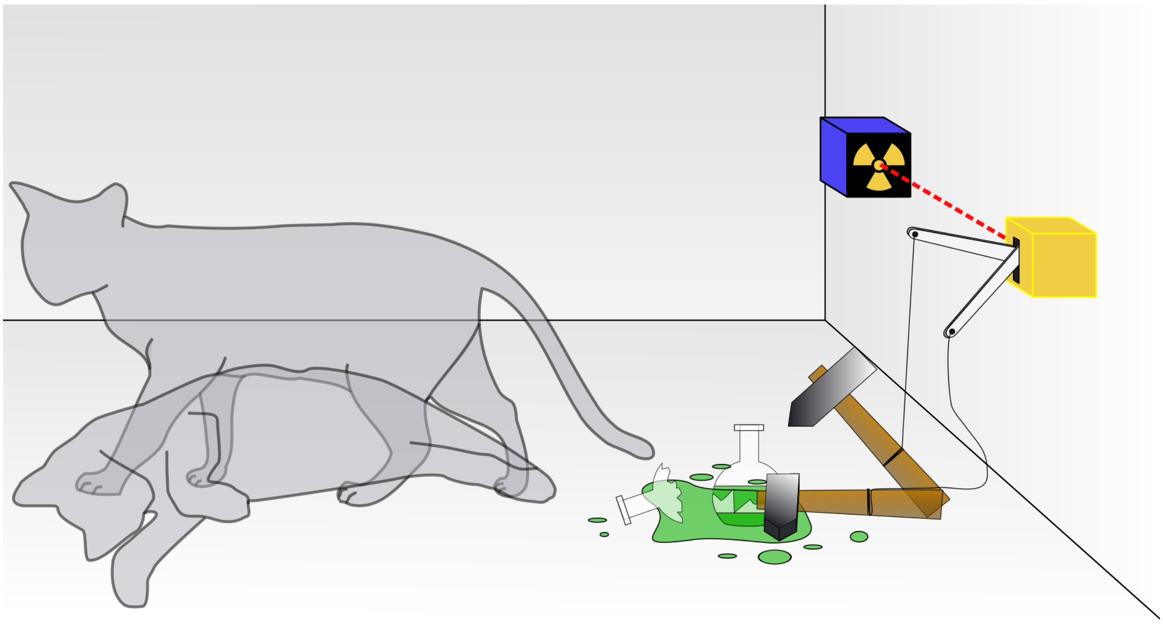
Refer to the slides given on this topic.

1. Consider a member of the column space. Suppose you were to describe it by an English word. Would that word be a noun, or a verb? Likewise, a member of the row space. (Write the respective description, and why you annotate it as a noun or verb.)
2. Are the configurations f and g on Slide 16 winnable? Provide evidence. If the configurations are winnable configurations, provide solutions in the form of a picture similar to the bigger picture in Slide 2.
3. Suppose you are given a configuration b (say the one in Slide 2) and the goal is NOT to turn the lights out, but instead, to get to a configuration b^*. For example, b^* may result in all anti-diagonals lit. Write three steps in solving this problem.

Your task

Please write or type the answers to the sub-questions above in a file called `Part1.pdf`. Make sure you write the answers in order. We are looking for an answer that fits on a page; write crisp answers.

Schrodinger's Cat



(From wikipedia)

The paradox offered by "Schrodinger's cat": Put a cat in a room in the presence of a lethal quantum event. If the Geiger counter detects radioactive decay, the glass is shattered, triggering a gas release. The wave function described in quantum mechanics suggests the cat exists in a superposition of dead/alive states simultaneously, only resolved when you open the room and make your observation. At that time, the cat is either alive or dead but not both alive and dead.

This put an emphasis on the observer that Einstein found objectionable.

In the sequel for this assignment, in addition to the cat, the room is also jinxed. Note: This assignment requires no understanding of Quantum Physics.

The Plot

One particularly amicable morning in Copenhagen, two old friends, Neils and Albert, were engaged in an interesting conversation about mysterious rooms and cats.

"You see my dear friend Albert, these rooms are quite out of the ordinary. We know with *absolute certainty* that there is a cat in each room. As you know these cats can only be in one of the three states — dead, alive or indeterminate.", said Neils rather matter-of-factly.

Albert knew that his friend was trying hard to perplex him but he played along. "Yes, I have been hearing a lot about these cats lately. They were discovered by some bloke named Erwin. What's so special about these rooms, then?" he said impassively, trying hard not to evince his sense of uneasiness.

"Well, according to Richard, who first *observed* their peculiar property, these rooms seemed to not obey any known law of physics. He said that whenever he used to open one of these rooms, the reality of the cat within it altered!" Neils paused for a dramatic moment but Albert was nonchalant. He continued, "if the cat was earlier known to be alive, it will be found dead when the room was opened. If he closed the door and opened it again, he would find the cat to be in an indeterminate state."

"Interesting indeed! And what did Richard observe if he opened the door of a room in which it was known that the cat is in an indeterminate state?" asked Albert, no longer able to hide his curiosity.

"The cat will be alive and kicking as if nothing had happened."

It all seemed rather *dicey* at first but soon he discovered that there was a consistent pattern to the way the rooms altered reality. At each observation, the cat in any room always *cycled* through these three states of existence (or realities) in the following order: ...alive, dead, indeterminate, alive, dead, indeterminate, alive, ..."

"So, if the cat was last found dead in a room, it will be in an indeterminate state next time the room's door is opened?" asked Albert, just to be sure.

"Yes," confirmed Neils.

"And if it was alive earlier, will it now be dead after opening the door?"

"Yes," reiterated Neils, "but don't get too surprised yet. He noticed something more.

You see, these rooms were arranged like this, here" continued Neils as he handed out a rough sketch of the layout of the mysterious rooms to Albert (See Figure 1).

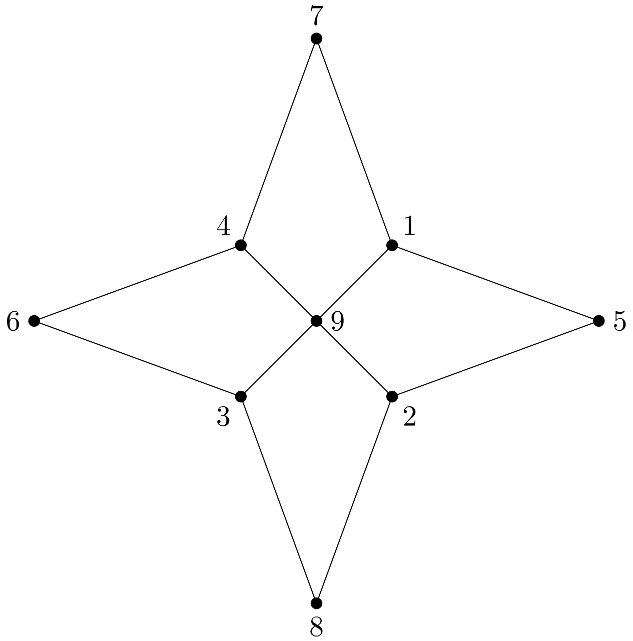


Figure 1: The star room

"You notice there are nine rooms here. Room *9* is adjacent to rooms *1,2,3,4*. Room *5* is adjacent to rooms *1* and *2*, and so on. He noticed that whenever he opened a room's door it also altered the realities in the adjacent rooms."

"You mean to say, that if I were to open and observe room *9*, not only will it alter the cat's state in room *9* but also in rooms *1,2,3* and *4*. If I were to observe room *5* it will impact the realities in rooms *5, 1* and *2*?" inquired Albert.

"Yes, that's right."

Deja Vu: Part 2

Last time Richard observed, the cats were in the following states in these rooms: 2, 1, 2, 0, 0, 1, 1, 0, 1 where his shortcut coding was 0: alive, 1: dead state, and 2: indeterminate. This configuration lists the states of the cats corresponding to room numbers 1, 2, 3, ..., and 9, respectively. We are interested in finding the minimum number of observations that he must make in order to ensure that all the cats are alive at the end of his observations. *Note that opening a door once (and noting the state of the cat within) accounts for one observation. The doors close automatically after each observation.*

Your task appears below.

Your task

Write a python code `simpleLayout.py` which solves the problem for the given layout of nine rooms for **any** given *configuration* of the cats. You may use the skeleton file `sample.py` provided to you to begin your code.

Input format

Input will be a space separated configuration of the cats in room 1, 2, 3, ... , 9 in order

Output format

Your program should print on the standard output, the minimum number of times Albert should open doors to ensure all the cats are alive finally. If he finds out that there is no way to make all the cats alive, print `Impossible State`.

In addition, if the configuration has a solution, write separately in a file the number of times each door should be opened as space separated vector for rooms 1, 2, 3, ..., 9 in order. The file name is `simpleLayoutSolution.txt`.

Example

Input: 2 1 2 0 0 1 1 0 1
Output displayed on screen: 6
Content of `simpleLayoutSolution.txt`: 0 0 0 0 0 2 2 0 2

The sequel to the sequel: Part 3

Having seen Albert solving the problem, he was presented with another one of Richard's problem with a different layout.

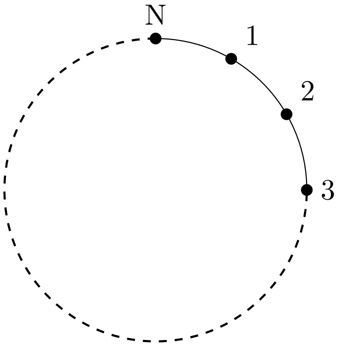


Figure 2

As you notice, here the rooms are all connected in a circular layout. Each room has exactly two immediately adjacent rooms neighbouring it. Each room is a meter away from its neighbours. As before, there is a cat behind each door and you are given the current configuration. As you open a door and observe a room, the state of the cat in it is altered according to the pattern mentioned earlier.

The catch this time, though is that as you open a door to observe a room, the reality of any room, which is exactly at a particular distance of f meters from the observed room, is also altered! Action at a distance!

Before you ask, f is any factor of N, the total number of rooms."

"So if I observe room x, then the reality of all rooms at $\pm f$ meters is also altered, where f is a factor of N. And, this happens for all factors of N?" asked Albert, to ensure he had understood the problem definition clearly.

"True indeed." "Now, don't look so worried. To make your task easier, assume that N is any number which can only have three factors. Neither more, nor less."

Your task

Write a python code `circleLayout.py` which solves the problem for N rooms for any given *configuration* of the cats in the circular set of rooms. Note that N is any number with exactly three factors.

You may use the skeleton file `sample.py` provided to you to begin your code.

Input format

First line in your input file will be N. Second will contain space separated configuration of cats in rooms 1, 2, ..., N in order

Output format

Your program should print the minimum number of times Albert should open doors to ensure all the cats are alive finally. If he finds out that there is no way to make all the cats alive print **Impossible State**. If the configuration has a solution write the number of times each door should be opened as space separated vector for rooms 1, 2, 3, ..., N in order in the solution which allows us to open minimum number of doors into a file named *circleLayoutSolution.txt*

Example

Input :
25
1 2 1 2 2 2 0 1 2 0 0 1 0 0 0 1 0 0 2 1 2 0 2 1 0
Corresponding Output displayed on screen :
11
Content of circleLayoutSolution.txt :
2 0 2 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 2 0 0 0 2 0

Submission Guidelines:

1. Do include a `readme.txt` (telling me whatever you want to tell me including any external help). Make sure to document individual percentages such as Student 1: 80%, Student 2:100%, Student 3:10%. In this example, the second student will get full marks (10/10) and the first student will receive 8/10. Sort by roll numbers. Do not forget the honor code.

2. `ReflectionEssay.pdf` will have multiple parts. First, make a commentary on what you learned from this assignment, and what someone who reads this at your academic level can learn. Next, provide the rationale behind solution provided by you for parts 2 and 3. There is no hard and fast rule on what is a perfect answer for this. Feel free to write less or more, but about half a page for each part is the minimum. Take extra effort in writing this since this will carry substantial marks.

3. Place `readme.txt`, `ReflectionEssay.pdf`, `simpleLayout.py` and `circleLayout.py` in one folder and compress it. The compressed version should be identifiable. For example, folder should be named `lab01_group07_final` and the related `tar.gz` should be named `lab01_group07_final.tar.gz`.

4. Please stick to .tar.gz. Do not use .tgz. Do use .zip

5. Your submission folder should look something like this:

```
   lab01_groupXY
└─ readme.txt
└─ ReflectionEssay.pdf
└─ PartI.pdf
└─ simpleLayout.py
└─ circleLayout.py
```