Cindy Zhou
DS210
Final Project
# Graph Coloring for Efficient Allocation of Radio Frequencies

## Introduction

How are radios able to communicate over long distances? Radio towers transmit signals through radio waves, which can then be received by other devices. To avoid interference with other radio towers, each radio tower will emit a specific frequency. However, due to the number of radio towers and the limited number of distinct frequencies, towers that are not close enough to interfere with each other can emit the same frequency. Some frequencies are better than others, so minimizing the number of frequencies used can improve communication. To solve this problem, graph coloring will be employed to figure out the least number of radio frequencies needed for a number of radio towers at a specific distance. This algorithm will be applied to a dataset of US radio towers.[1]

## Research & Data Preparation

To understand how to tackle this problem, it is necessary to understand the constraints of radio towers. Radio towers and distances between them will be represented by nodes and edges respectively. Because the distance is the same between two towers, this can be an undirected graph. Since this dataset includes towers all over the United States, it is necessary to define what an adjacent tower is. FM radio reception distances are usually 50 km, so towers that are 100 km away from each other will not interfere or have any overlap.[2] As such, they can be given the same signal, or colored the same way, and will not need an edge between them.

Since only the distance and a way to keep track of which tower is which, the data file contains a lot of extraneous information. To find the distance only the latitude and longitude are needed, which can be done using the Haversine formula. All columns besides id, latitude, and longitude can be removed. Data was cleaned to make sure points were latitude and longitude were valid floats.

## Code Walkthrough

Main first calls to functions in module read_csv, which takes care of all csv processing. The code first checks to see if the file is valid using serde, exiting the process if not. The csv file is then read into a vector of tuples (id, latitude, and longitude) using the Bufreader crate to read the csv line by line. This vector is returned, and serves as the data used for the rest of the code.

Variable called limit is then used to set the maximum distance of an edge. As research above, this limit is 100 km. With the number of nodes (data.len()), data, and limit, it is now possible to make a graph, which stores the distance between two points as an edge. All functions are in the module graph_maker. I chose to use an adjacency matrix in the case that the graph was dense, since some towers are from different companies, initializing all values of distance as 0.0, since radio towers cannot be on top of each other. The distance function uses the latitude and longitude to calculate the distance between two nodes through the Haversine formula, converting degrees into km by using the radius of the earth and trigonometry. If the distance exceeds the 100 km limit, no edge needs to be filled in since these nodes can be colored the same. The fill_matrix function puts this value into the correct space in the graph matrix. This is then returned as the variable graph.

Lastly, the coloring algorithm is used to color each node in the graph. Colors represent unique radio frequencies. The graph coloring algorithm is greedy, starting by filling the first node with one color and filling in the next nodes with any available color. To do this, the code keeps track of a list of colors present in the graph. Each vertex implements a variable called unavailable, which finds all the previous colors used by adjacent nodes. If the list of colors matches unavailable, meaning all colors are taken already, a new color is used. Else, the node will take the first color that is not in unavailable. The code returns the colored nodes, as well as the number of frequencies that would be needed.

## Data Analysis

The graph shows that for the 19938 radio towers in the US, only 113 radio frequencies need to be used. This is effective, considering this is 176 times less frequencies needed, clearing up the airways for other parties that need radio signals. Radio towers in the US operate from 88 to 108 mHz (TV takes 82-88 mHz and aeronautical signals take 108-136 mHz), and each signal is about 0.2 mHz wide because of the bandwidth of the data transmitted.[2] As such, there are 100 possible unique frequencies, which indicates this algorithm is fairly close, but could be better.

It is important to note that this work utilizes the greedy algorithm, one of the faster but less optimal graph coloring algorithms. The coloring algorithm alone has a runtime of O(n), and the whole algorithm is slowest when filling in the matrix, which has a runtime of O(n^2). Considering the solution is fairly close to actual FM radio frequencies and the speed of the algorithm, this is a quick way to allocate frequencies for a vast number of towers in an efficient way. More advanced algorithms will be necessary to more efficiently color these graphs, but for these purposes the greedy algorithm does fairly well in conserving the number of frequencies radio towers need.

## Works Cited

1. OP, M. (2022, May 28). *Cellular towers in the United States*. Kaggle. Retrieved December 12, 2022, from https://www.kaggle.com/datasets/mattop/cellular-towers-in-the-united-states?resource=download
2. *How much distance does a radio transmitter cover?* High End FM Transmitters and Professional Equipment for Radio Stations | Teko Broadcast. (2021, October 28). Retrieved December 12, 2022, from https://www.tekobroadcast.com/en/blog/how-much-distance-does-radio-transmitter-cover
3. *Why do FM frequencies end in an odd decimal?* Federal Communications Commission. (2021, September 1). Retrieved December 12, 2022, from https://www.fcc.gov/media/radio/fm-frequencies-end-odd-decimal#:~:text=The%20FM%20broadcast%20in%20the,lower%20end%20of%20the%20channel.