Input: Artist

Make a graph of related artists from that artist. From that artist, find Spotify's 20 related artists and add to the graph, then for each of those 20 artists, find their 20 related artists and add to the graph, etc. (can limit the size of the graph based on runtime feasibility, API call limit. Maybe stick to 5 degrees of separation from the original artist?)

Do something interesting with that graph (cycle detection, idk)

Output:

Tasks: Find out how Spotify API works
- C++ API library
- C++ JSON parser
- Email the TAs or whatever about this project

Communication:

1. Team Meetings When and how often often will your team meet? How long should each meeting last? What software or tool will you use to host these meetings? Will someone take notes (record minutes)?

- **Meeting 2-3 times a week through Zoom or Facetime. Evan Matthews volunteers to take notes (record minutes) for team progress and items to be done in the following weeks.**

2. Assistance How will your teammates be able to contact you if they need your help or opinion on a task? How quickly should you be expected to respond?

- **Teammates can contact each other through a group chat on iMessage; preferably, each teammate should check this chat daily and respond to messages the same day, but one or two rare responses of 24-48 hours are understood.**

3. Respect An effective team needs to have an environment which encourages open expression of ideas. How will you ensure that every member has an opportunity to speak and, more importantly, that every member will actively listen and engage with the thoughts of others?

- **Teammates are expected to give everyone an opportunity to speak and/or give suggestions during meetings. Before pursuing action, important decisions regarding the project should be agreed upon by ALL teammates.**

Collaboration:

1. Work Distribution How will you assign workload for this project? How will you address unexpected complications or unforeseen work? You are encouraged to identify the strengths and desires of each team member when distributing work. You do not need to all work equally on a particular deliverable – it is the overall work that should be largely equal.

- **Everyone will be working on the code. Cale and Kevin will be responsible for coding the access to Spotify's API. Beth and Evan will be responsible for the graph implementation. The entire team will be working on the solution to find a playlist that connects two artists. The README will be spearheaded by Kevin, and reviewed by the team members. Meeting minutes will be kept track of by Evan which will be part of our final presentation. Cale and Beth will start the final presentation while the entire team is expected to make contributions.**

2. Time Commitment How many hours of work per week is expected of each group member? Are there prior time commitments that need to be accounted for? How will you address conflicts or commitments when they do occur?

- **Team members are expected to commit ~10 hours of work towards the project per week. This should be similar to an MP. Kevin currently does not have any commitments during the week outside of classes in the morning**

and paper for a philosophy class. Cale has no time commitments beyond classwork, including a final project for CS361. Evan will be editing a podcast episode through December, which will be done concurrently with the final project. Beth has a part-time job (20 hr/week) and will not be able to meet with the team or work on the project when she is working. She also has the final project for CS 361. To try and account for everyone, we will have most of the work done asynchronously. The meetings are to report new information or research, or a status update.

3. Conflict Resolution How will the team resolve situations where there is a disagreement between members? Situations where one or more members have not accomplished their tasks? Situations where one or more members are habitually late? You are encouraged to bring such issues to course staff, but only after first trying to resolve the issue yourself.

- **If there is a disagreement between members then we will confront the team member. Kevin is not afraid to directly speak to the team member and encourages the other members to speak to him directly if there is a disagreement. If there are disagreements between ideas then a meeting will be held to form a compromise. If there are issues with work then again, a meeting will be held between all members to address the situation and make sure the team member is doing their fair share of work.**

Project Proposal:

Kevin Zhou kjzhou2, Beth Fineberg erf3, Cale Wolf caleqw2, Evan Matthews evanmm3

Our project will be dealing with artists on Spotify and their connections with each other. We will be using the Spotify API, and we will use HTTP GET requests to retrieve the data. We understand that we will basically be building our own database, different from the prepared datasets of OpenFlights or the SNAP libraries. In this project, we will be able to input an artist able to be found on spotify, and give information about the artists that they are connected to. In each album that a main artist produces, it also lists the co-artists. This is how we will be building our graph of artist connections. There will not be weights for any of the connections. To access the API, we will be using an HTTP Protocol using cURL. This may change depending on whether we are able to access the package but we would use either the cpr or curlpp packages to send HTTP requests.The data that Spotify will be returning to us will be a JSON object, so we will have to either build our own JSON parser or use nlohmann::json from github to parse the data. With the ability to use GET requests and parse JSON files, we would build a graph of Spotify artists linked to each other. With this data, we will be able to create a playlist that connects any two artists with songs that artists have collaborated on. This would be similar to the shortest path algorithm for the OpenFlights data. Limitations in this project is the speed at which we are able to access the API. With multiple requests, the Spotify API is rate-limiting, which means that we would have to retry the requests after some time if we send too many. This can really slow down our search, especially considering the number of artists that are on the platform. If this does happen, we may be able to remedy this by only selecting the first 20 artists that the main artist has collaborated with. With less connections, there are less calls to the API and could possibly speed up our program.

Which artists are n degrees away from a specific artist on spotify?

How to answer: Create a graph of (edges: artists, vertices: album that both artists worked on)

OR shortest path between two artists based on related albums.