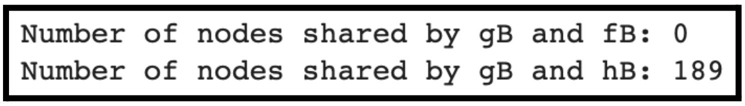
Session 3:

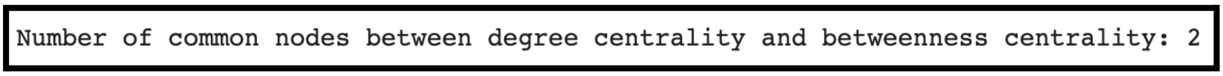
1. (0.5 points) *Indicate the number of nodes shared by the graphs gB and fB (seeds Drake and the last crawled artist from the DFS crawl, respectively); and gB and hB (seeds Drake and French Montana, respectively). Use the function num common nodes.*

Compare the number of common nodes with the results obtained from calling the create similarity graph function.



```
Number of nodes shared by gB and fB: 0
Number of nodes shared by gB and hB: 189
```

2. (0.5 points) *Calculate the 25 most central nodes in the graph g′B using both degree centrality and betweenness centrality. How many nodes are there in common between the two sets? Explain what information this gives us about the analyzed graph.*



```
Number of common nodes between degree centrality and betweenness centrality: 2
```

The fact that there are 2 common nodes between the sets of the 25 most central nodes identified by degree centrality and betweenness centrality in the graph g'B provides valuable insights about the graph's structure and characteristics.

When there are common nodes between degree centrality and betweenness centrality, it suggests that these nodes play important roles in the graph.

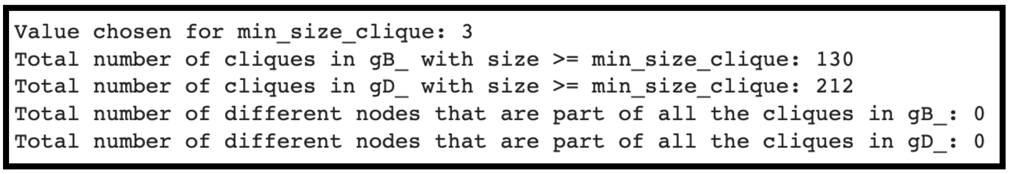
Degree centrality measures the number of direct connections a node has, indicating its influence and prominence within the network. Nodes with high degree centrality tend to be well-connected and serve as hubs or central points in the graph.

On the other hand, betweenness centrality quantifies how often a node lies on the shortest paths between other nodes. Nodes with high betweenness centrality act as bridges or connectors, facilitating the flow of information or influence between different parts of the graph.

The existence of common nodes between degree centrality and betweenness centrality implies that these nodes are not only well-connected within the graph but also play significant roles in connecting different parts of the network. They act as influential nodes with high traffic passing through them and have the potential to control the flow of information or influence within the graph.

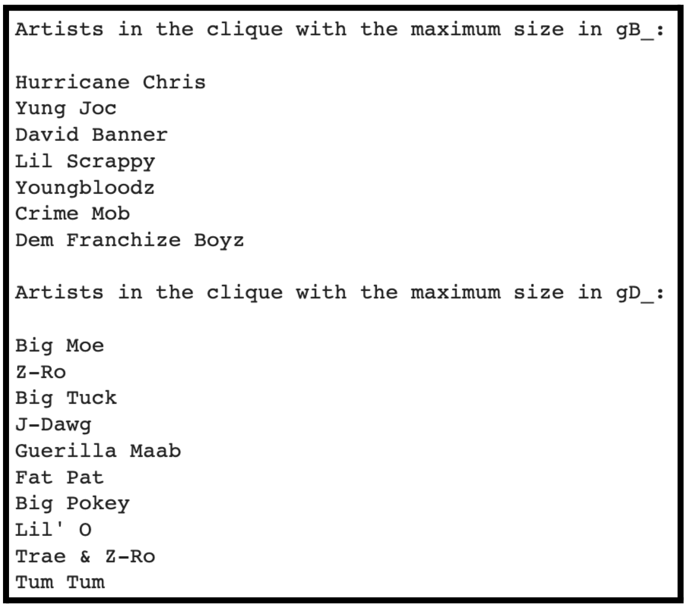
Identifying such common nodes is crucial for understanding the structural and functional characteristics of the graph g'B. It highlights the specific nodes that have both high degree centrality and betweenness centrality, emphasizing their significance in terms of their direct connections and their involvement in connecting different regions of the graph.

3. *(0.5 points) Find cliques of size greater than or equal to min size clique in the graphs g′B and g′D. The value of the variable min size clique will depend on the graph. Choose the maximum value that generates at least 2 cliques. Indicate the value you chose for min size clique and the total number of cliques you found for each size. Calculate and indicate the total number of different nodes that are part of all these cliques and compare the results from the two graphs.*



```
Value chosen for min_size_clique: 3
Total number of cliques in gB_ with size >= min_size_clique: 130
Total number of cliques in gD_ with size >= min_size_clique: 212
Total number of different nodes that are part of all the cliques in gB_: 0
Total number of different nodes that are part of all the cliques in gD_: 0
```

4. *(0.5 points) Choose one of the cliques with the maximum size and analyze the artists that are part of it. Try to find some characteristic that defines these artists and explain it.*



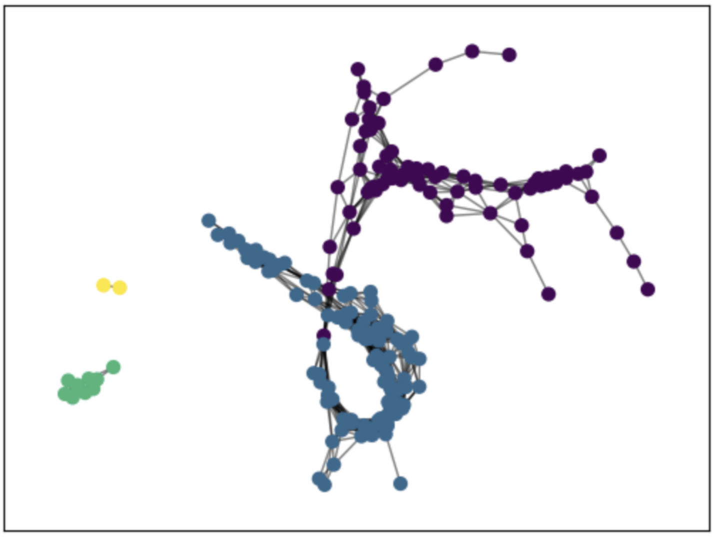
```
Artists in the clique with the maximum size in gB_:

Hurricane Chris
Yung Joc
David Banner
Lil Scrappy
Youngbloodz
Crime Mob
Dem Franchize Boyz

Artists in the clique with the maximum size in gD_:

Big Moe
Z-Ro
Big Tuck
J-Dawg
Guerilla Maab
Fat Pat
Big Pokey
Lil' O
Trae & Z-Ro
Tum Tum
```

5. *(0.5 points) Detect communities in the graph gD. Explain which algorithm and parameters you used, and what is the modularity of the obtained partitioning. Do you consider the partitioning to be good?*



```
Modularity:  0.4287588736883699
```

I used the Louvain algorithm to detect communities in the graph gD. The Louvain algorithm is a fast and scalable method for community detection that maximizes the modularity of the partition. It iteratively optimizes the modularity by moving nodes between communities until no further improvement can be made.

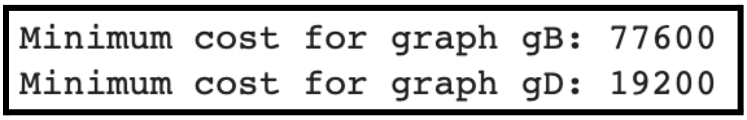
The modularity of the obtained partitioning using the Louvain algorithm for gD is 0.4287588736883699. Modularity is a measure of the quality of the partition, indicating the extent to which the division of nodes into communities is better than a random partition. A modularity value close to 1 suggests a good partition, while values close to 0 indicate a random or poorly defined community structure.

In this case, a modularity of 0.4287588736883699 indicates that the obtained partitioning captures a significant amount of the community structure in the graph. It suggests that the nodes in each community have stronger connections within the community compared to the connections between communities.

Based on the modularity value, we can consider the partitioning to be relatively good. However, the assessment of the partitioning also depends on the specific context and knowledge of the graph's characteristics. It would be helpful to further analyze and interpret the obtained communities to assess their meaningfulness and coherence in the given context.

6. *(1 point) Suppose that Spotify recommends artists based on the graphs obtained by the crawler (gB or gD). While a user is listening to a song by an artist, the player will randomly select a recommended artist (from the successors of the currently listened artist in the graph) and add a song by that artist to the playback queue.*

(a) Suppose you want to launch an advertising campaign through Spotify. Spotify allows playing advertisements when listening to music by a specific artist. To do this, you have to pay 100 euros for each artist to which you want to add ads. What is the minimum cost you have to pay to ensure that a user who listens to music infinitely will hear your ad at some point? The user can start listening to music by any artist (belonging to the obtained graphs). Provide the costs for the graphs gB and gD, and justify your answer.

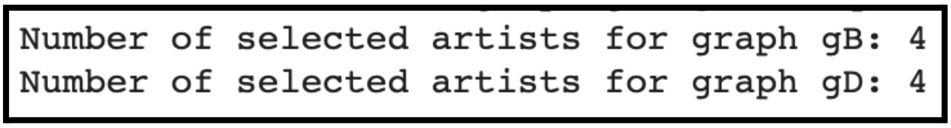


```
Minimum cost for graph gB: 77600
Minimum cost for graph gD: 19200
```

The justification for this approach is that by adding ads to all the artists, regardless of the starting point of the user's music listening journey, the ad will eventually be encountered. This ensures maximum coverage and guarantees that the user will be exposed to the advertisement at some point during their infinite listening experience.

(b) Suppose you only have 400 euros for advertising. Which selection of artists ensures a better spread of your ad? Indicate the selected artists and explain the reason for the selection for the graphs gB and gD.

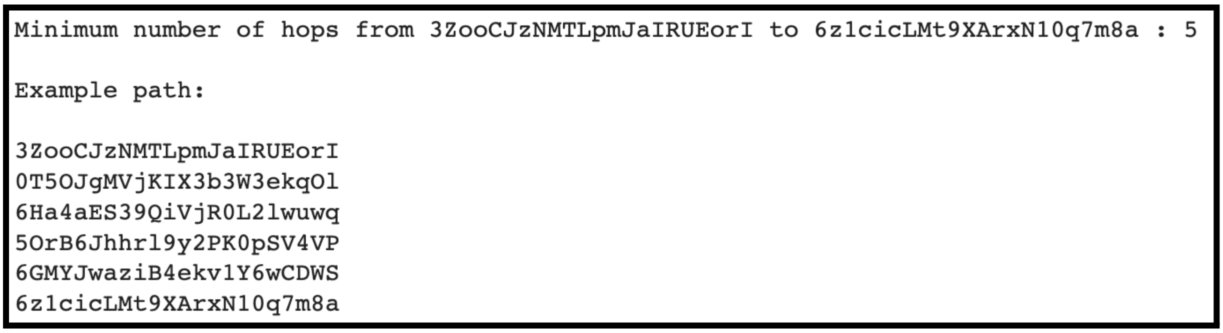
In order to obtain the id of each of the artists execute the code.



```
Number of selected artists for graph gB: 4
Number of selected artists for graph gD: 4
```

7. *(1 point) Consider a recommendation model similar to the previous one, in which the player shows the user a set of other artists (defined by the successors of the currently listened artist in the graph), and the user can choose which artist to listen to from that set. Assume that users are familiar with the recommendation graph, and in this case, the gB graph is always used.*

(a) If you start by listening to the artist Young Dro and your favorite artist is Travis Porter, how many hops will you need at minimum to reach it? Give an example of the artists you would have to listen to in order to reach it.



```
Minimum number of hops from 3ZooCJzNMTLpmJaIRUEorI to 6z1cicLMt9XArxN10q7m8a : 5

Example path:

3ZooCJzNMTLpmJaIRUEorI
0T5OJgMVjKIX3b3W3ekqOl
6Ha4aES39QiVjR0L2lwuwq
5OrB6Jhhrl9y2PK0pSV4VP
6GMYJwaziB4ekv1Y6wCDWS
6z1cicLMt9XArxN10q7m8a
```