

# Blending Two Automatic Playlist Generation Algorithms

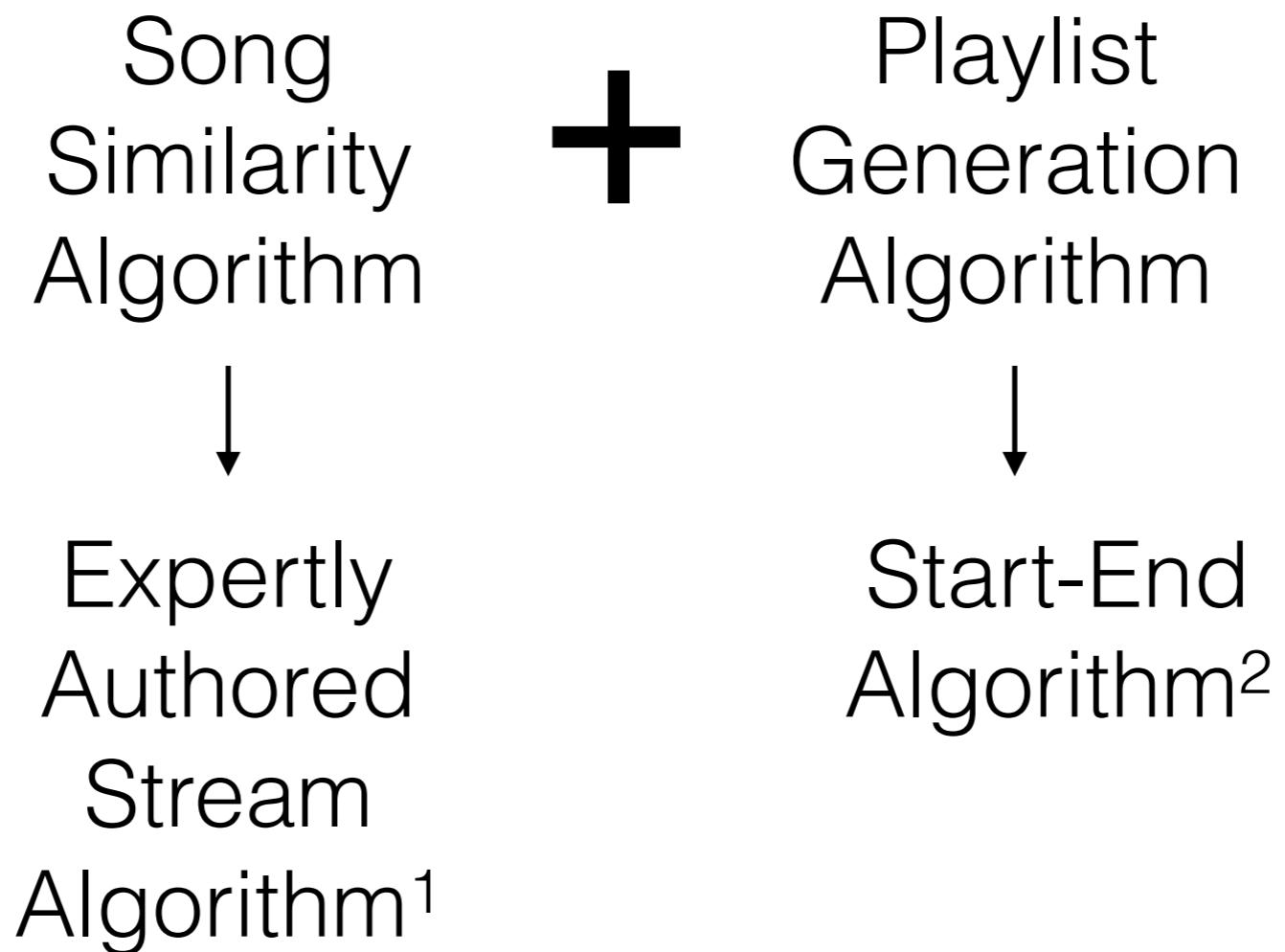
James Curbow

Advisor: Matt Anderson

# Background

Song  
Similarity  
Algorithm + Playlist  
Generation  
Algorithm

# Background

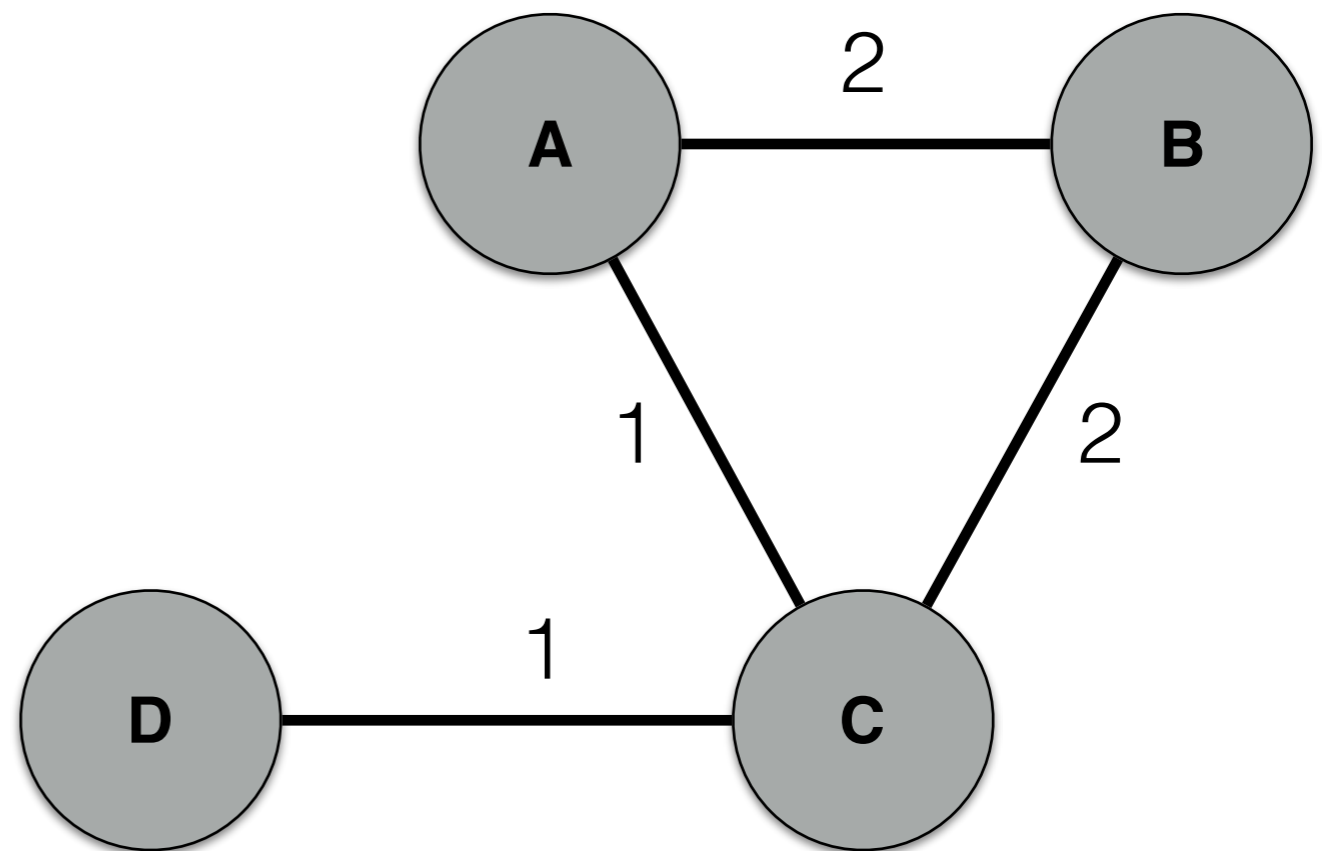


# Research Question

*“Will this generate smooth playlists? Can we improve either algorithm?”*

# Expertly Authored Stream Graphs<sup>1</sup>

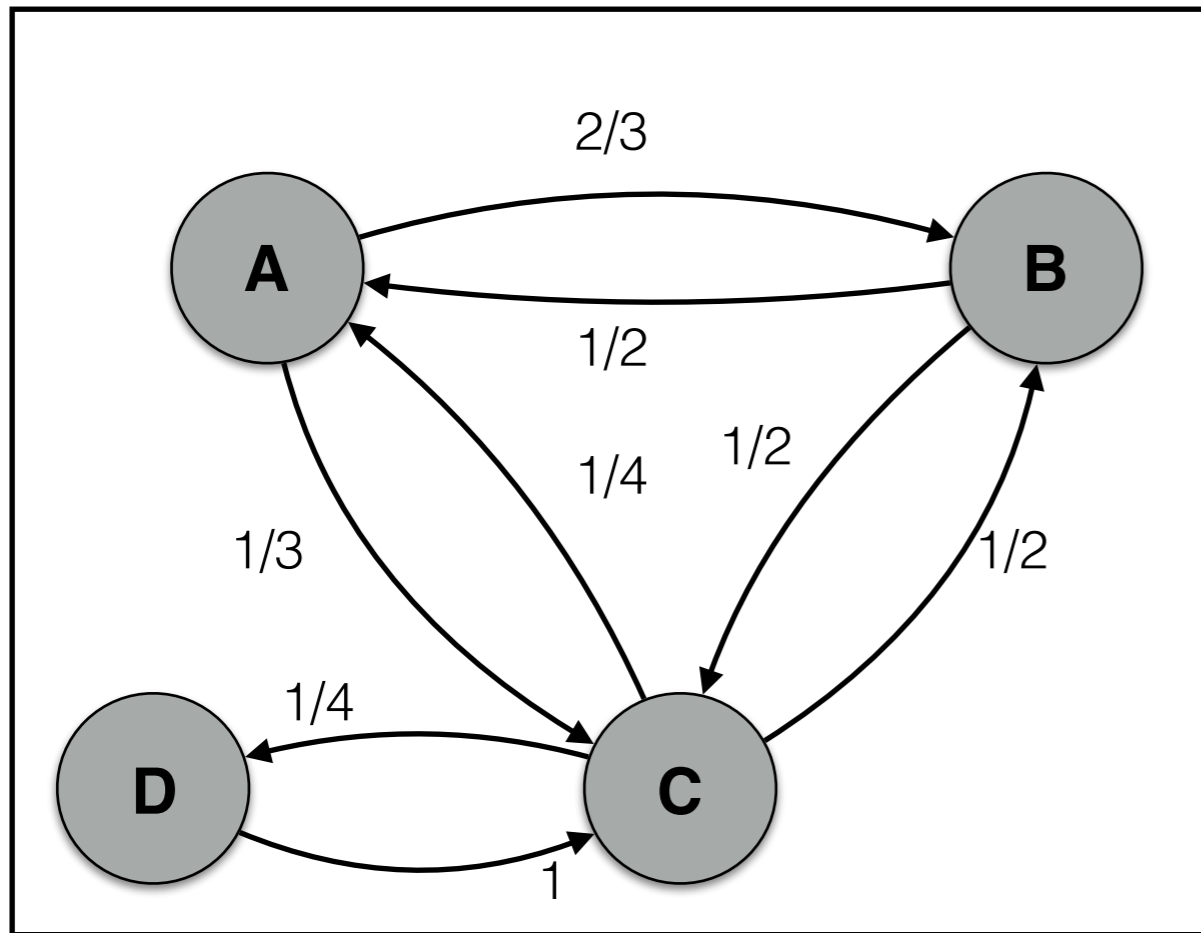
Playlist	
1	Song A
2	Song B
3	Song C
4	Song B
5	Song A
6	Song C
7	Song D



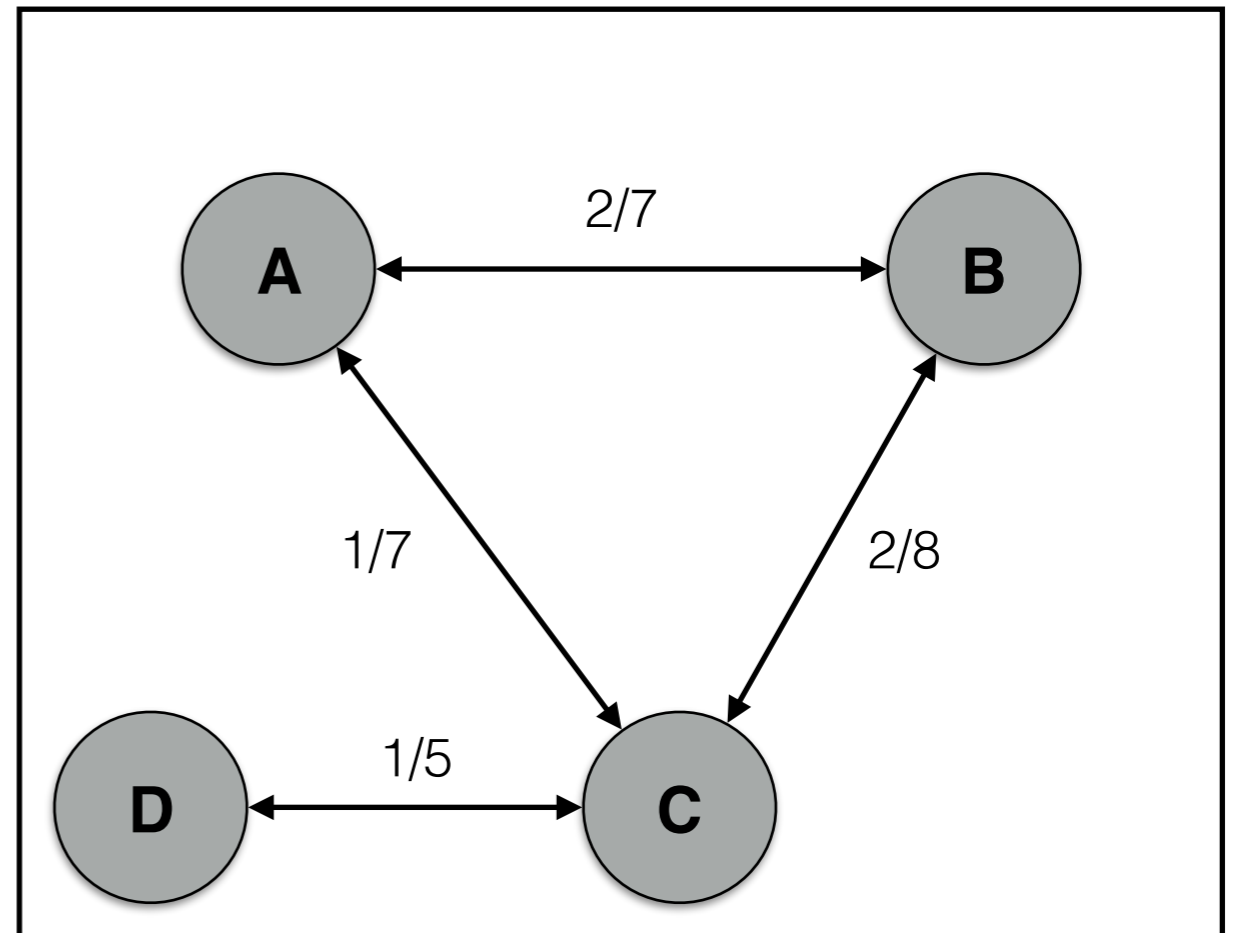
[1]R. Ragno, C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *Proceedings of the 7th ACM SIGMM international workshop on MIR*, pages 73–80. ACM, 2005.

# Similarities

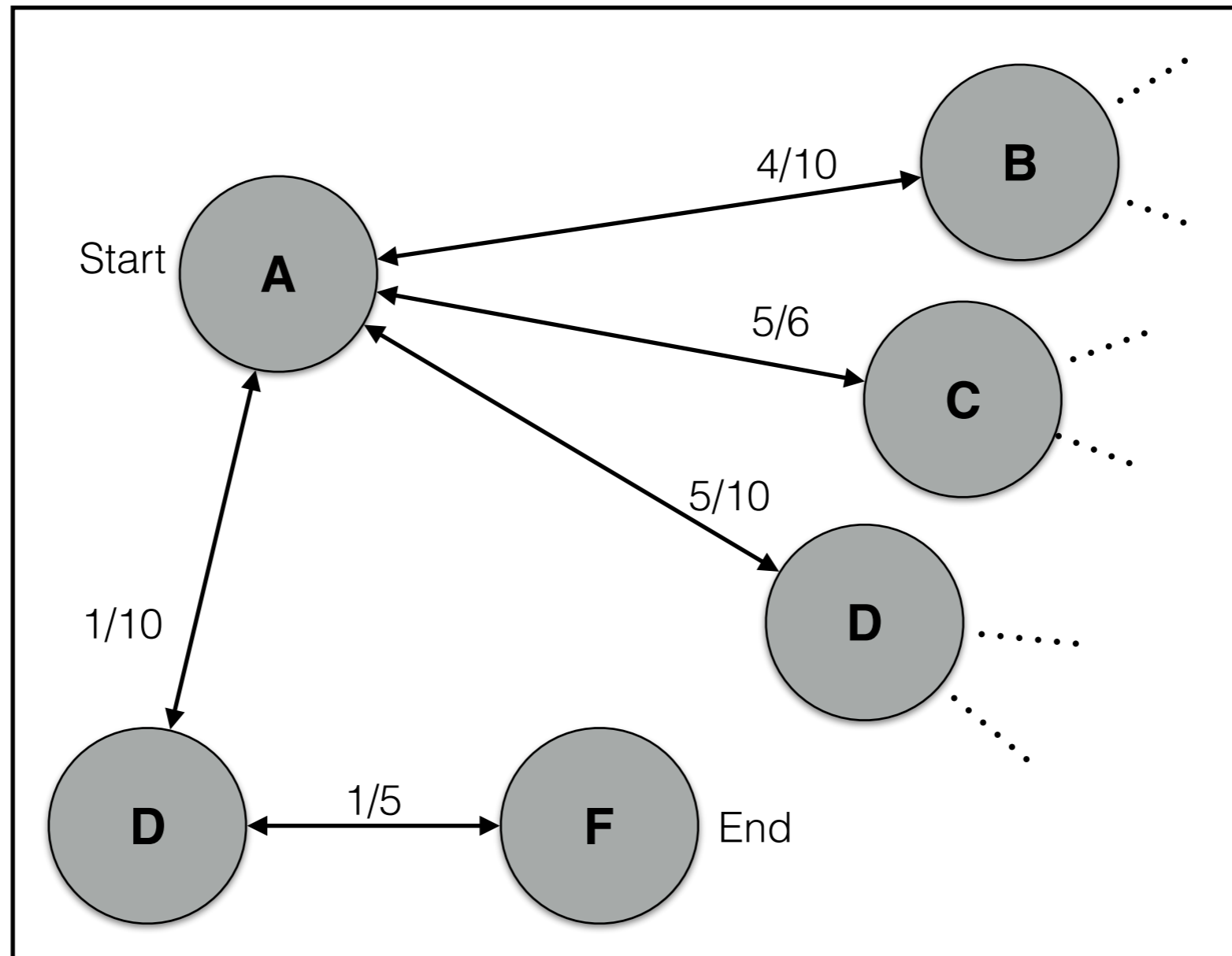
One Way



Two Way

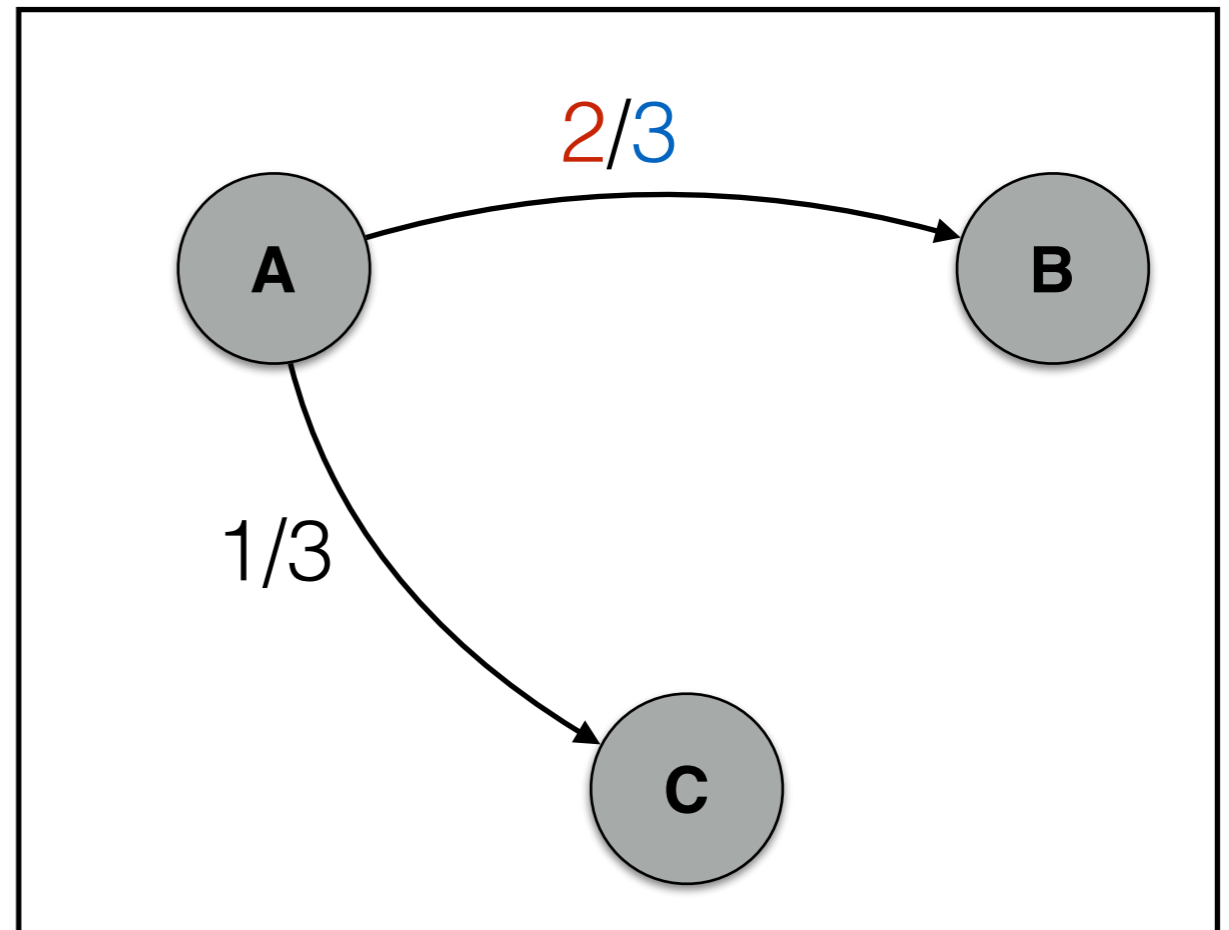
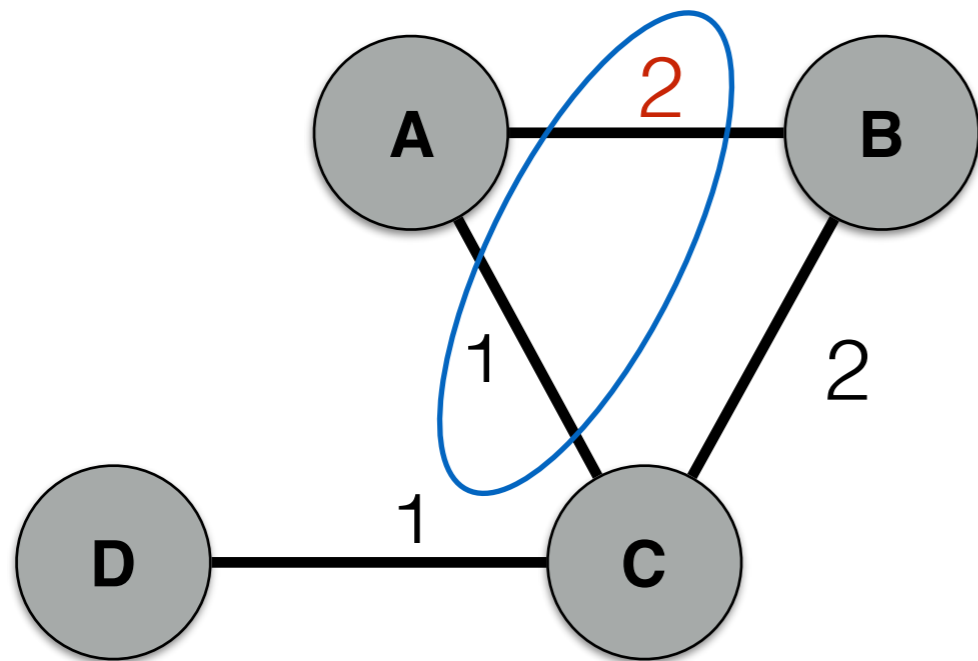


# Two Way



# Similarities (example)

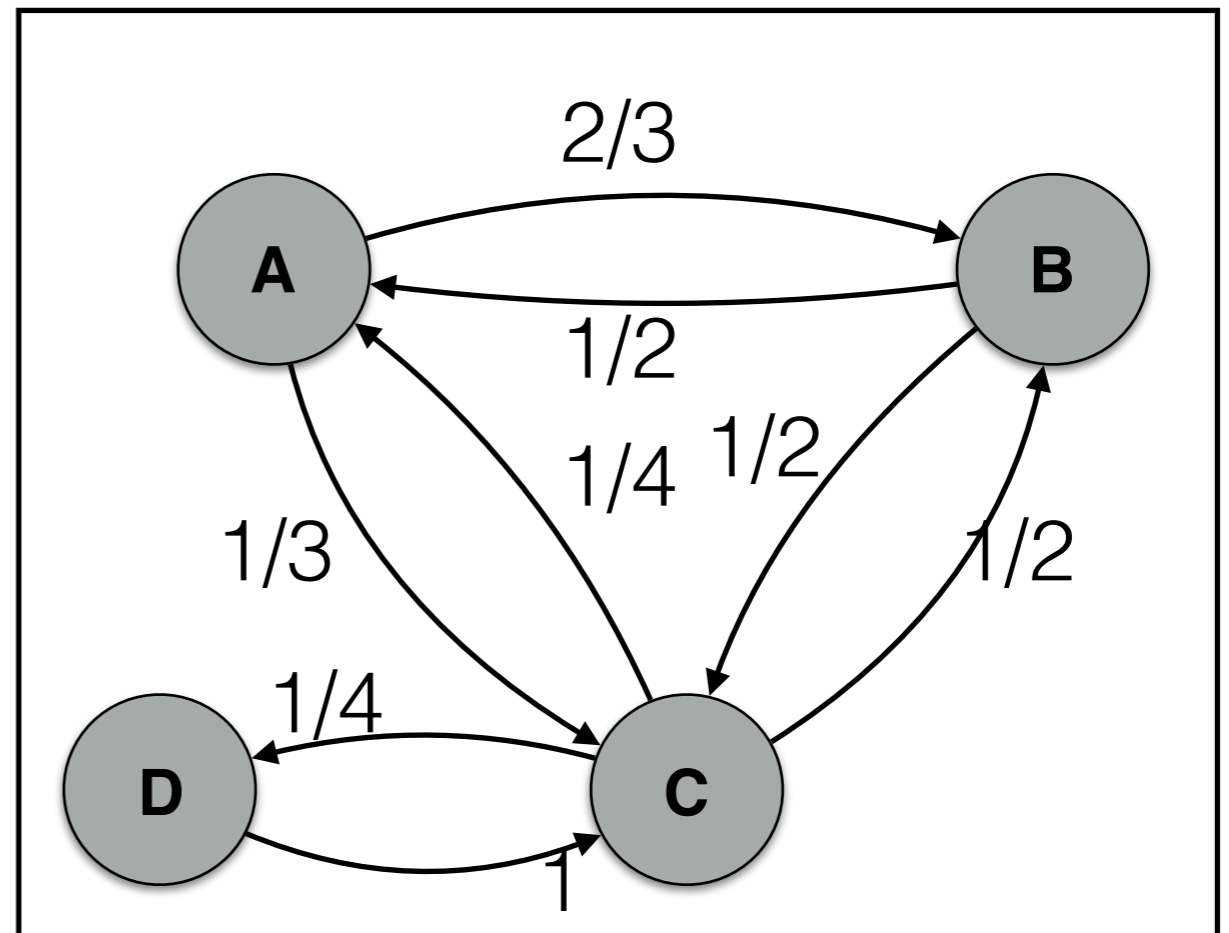
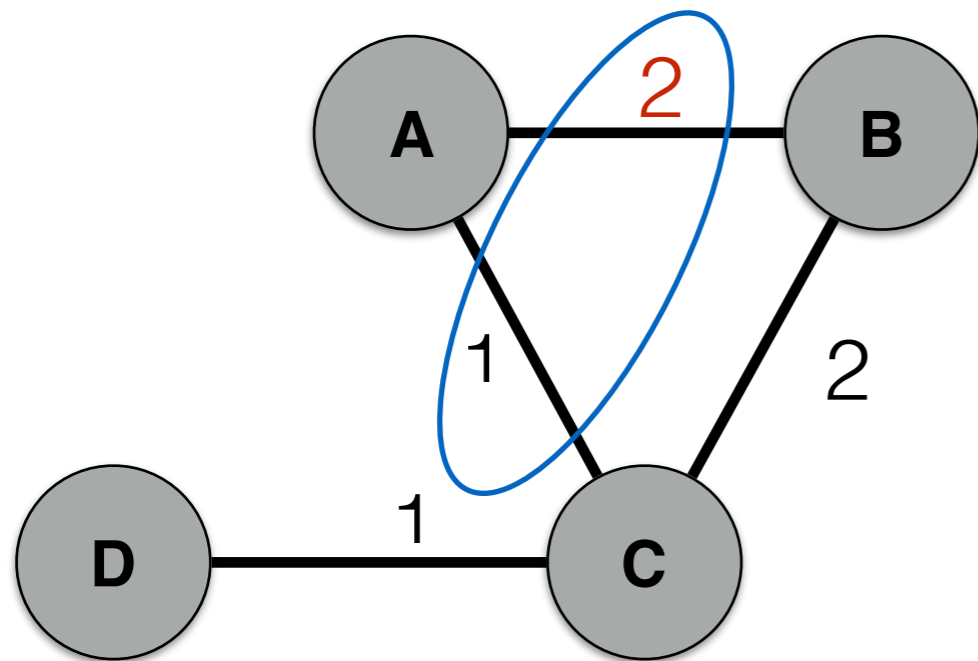
One Way





# Similarities (example)

One Way



# Data

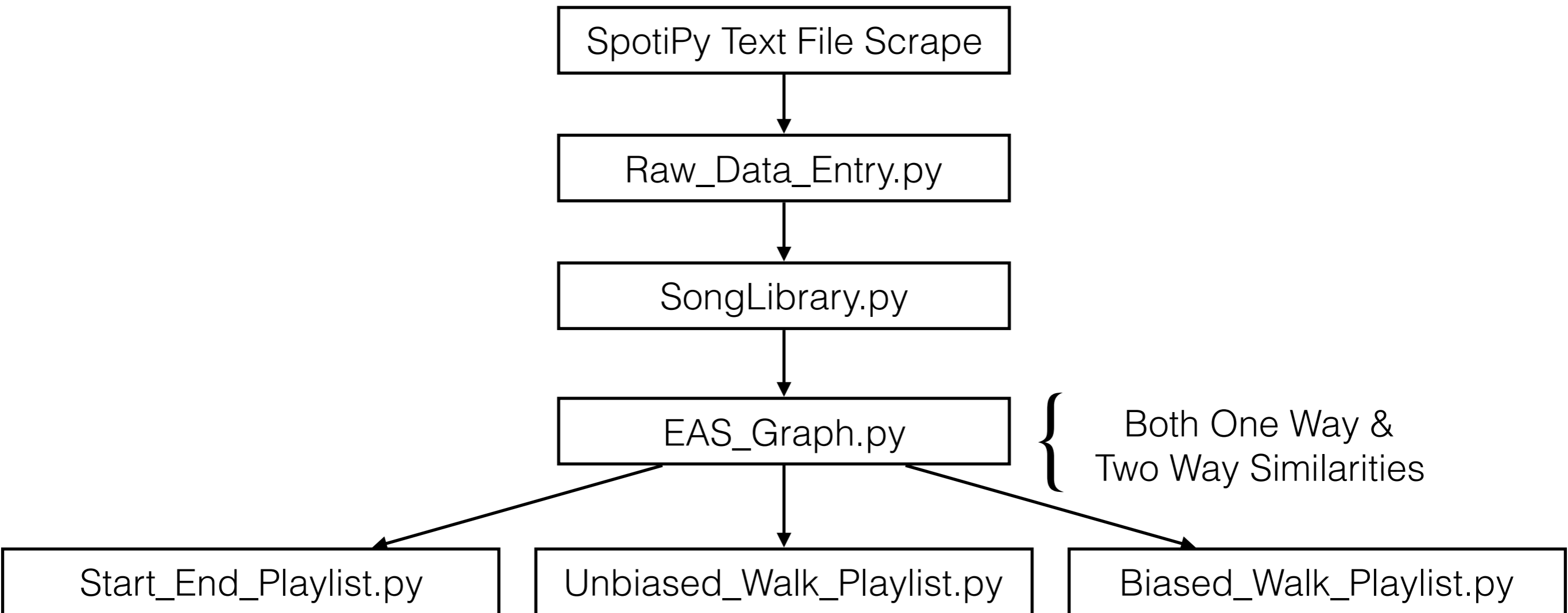
- Spotify Web API + SpotiPy
  - ~250,000 Songs
  - ~2,600 Expertly Curated Playlists



+



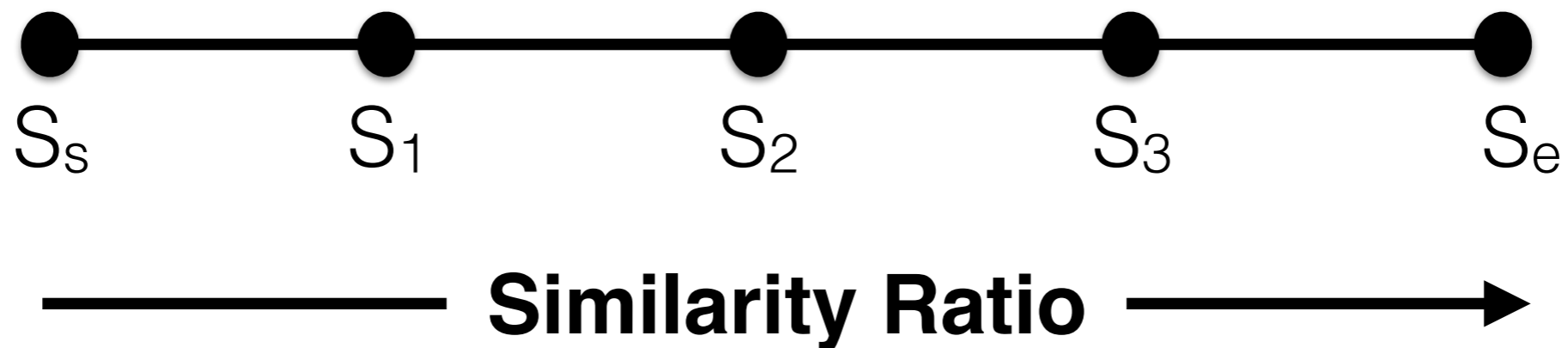
# Implementation



# The Start-End Song Algorithm<sup>1</sup>

- Picks and orders songs based on **increasing Similarity Ratios**

**Similarity Ratio:**  $\frac{\text{sim}(S_i, S_e)}{\text{sim}(S_i, S_s)}$

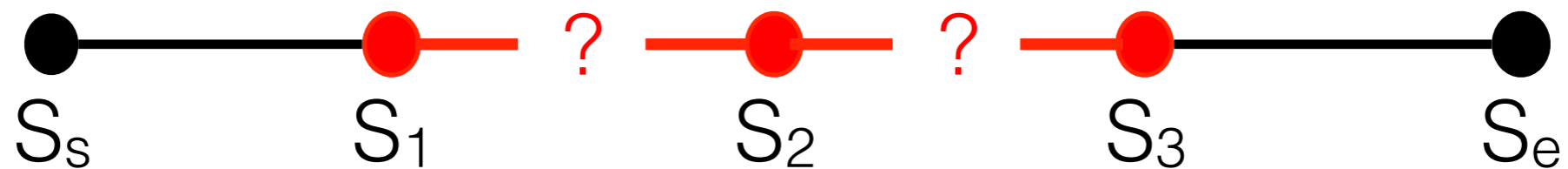


[1]A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *ISMIR*, pages 173–178, 2008.

# The Start-End Song Algorithm<sup>1</sup>

- Picks and orders songs based on increasing **Similarity Ratios**

**Similarity Ratio:**  $\frac{\text{sim}(S_i, S_e)}{\text{sim}(S_i, S_s)}$



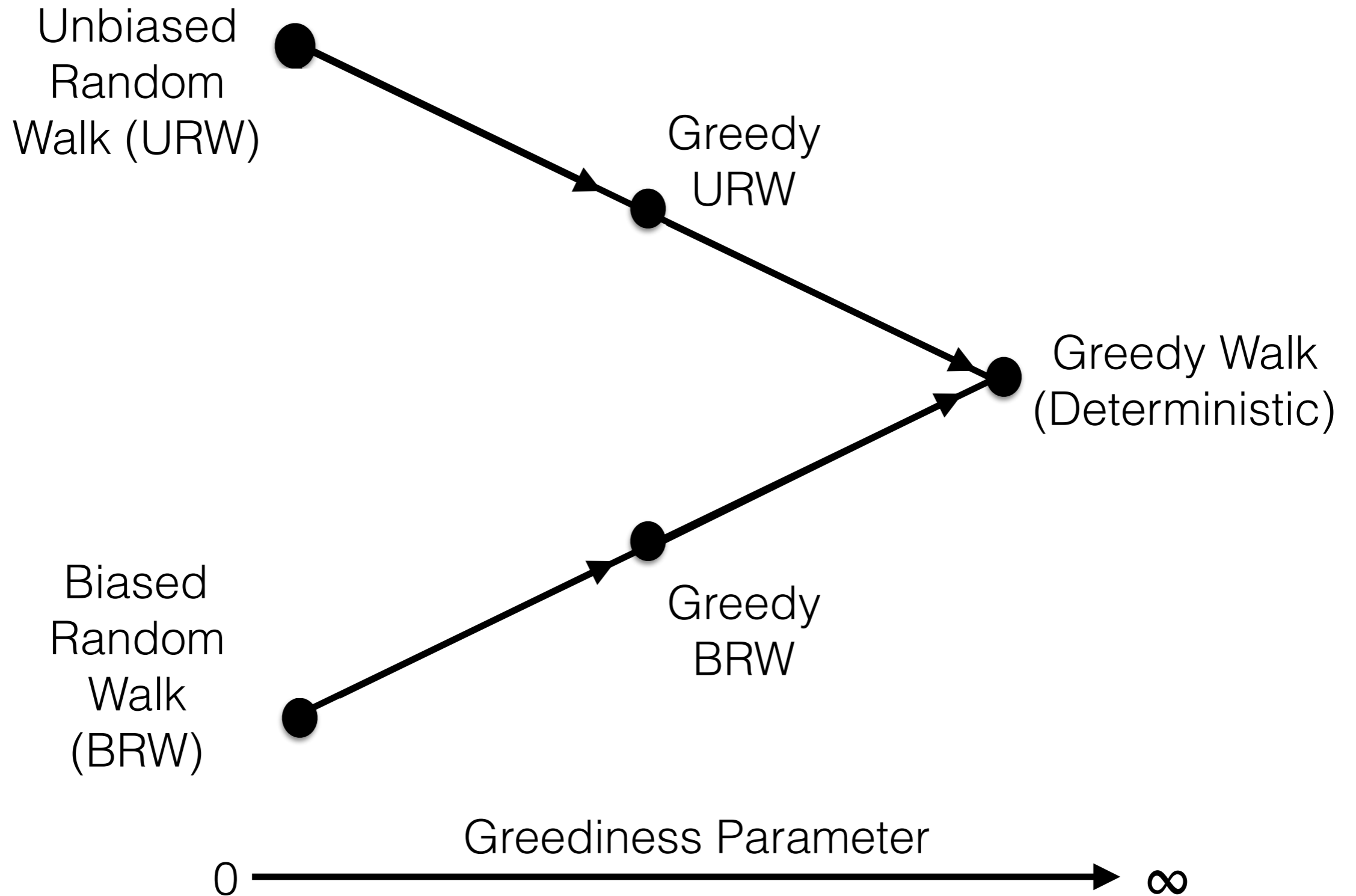
[1]A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *ISMIR*, pages 173–178, 2008.

# Start-End Algorithm Results

## Issues:

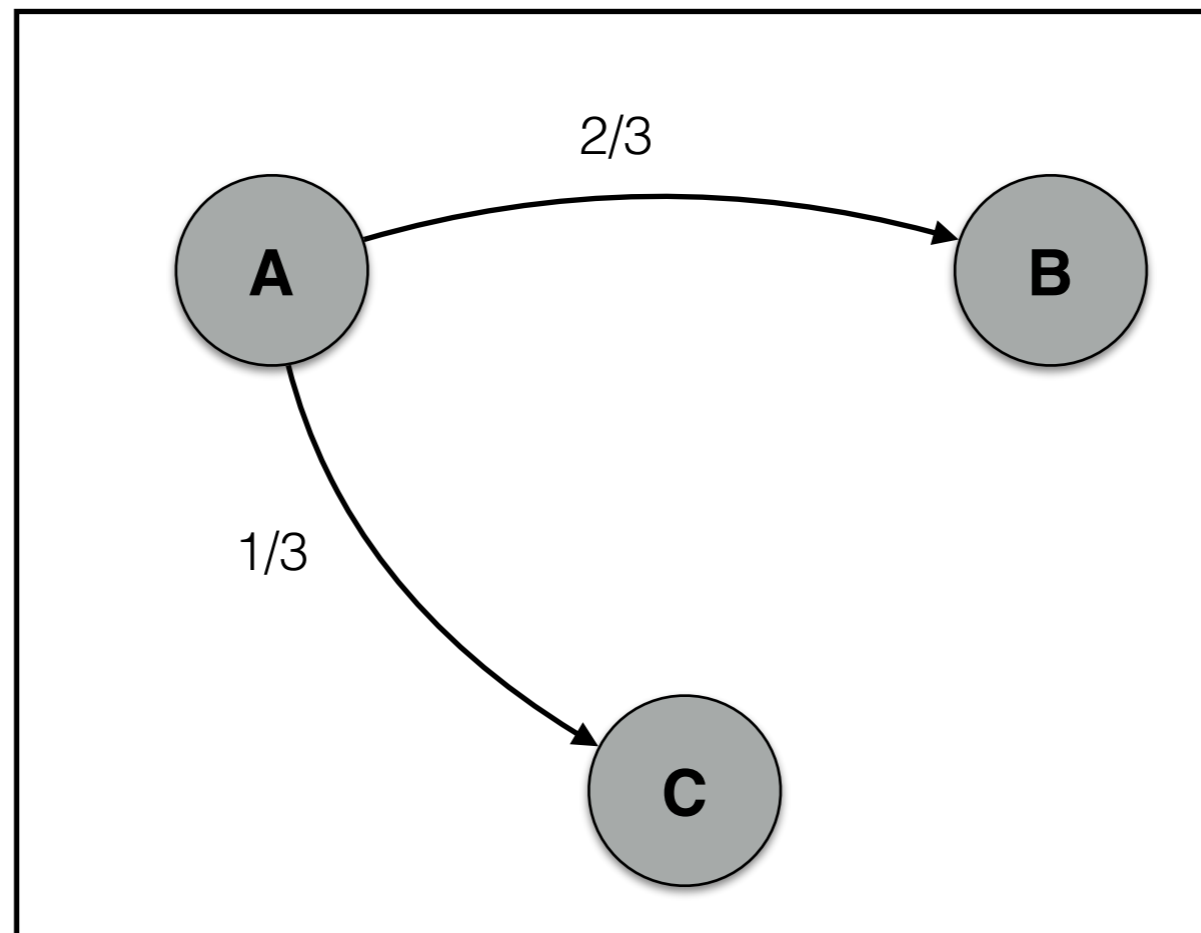
- 1) Adjacent similarities are never ensured...
- 2) Every song **needs** to have non-zero similarity to start & end songs...
- 3) Unclear assumptions about similarity definition...

# New Algorithms



# Similarities

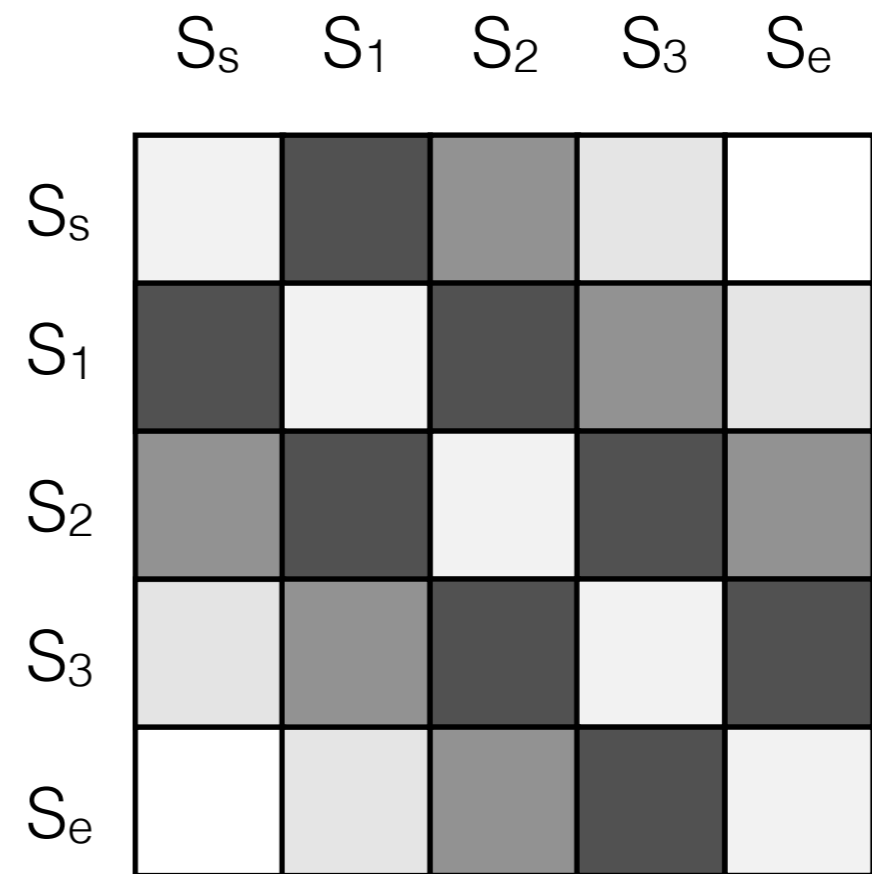
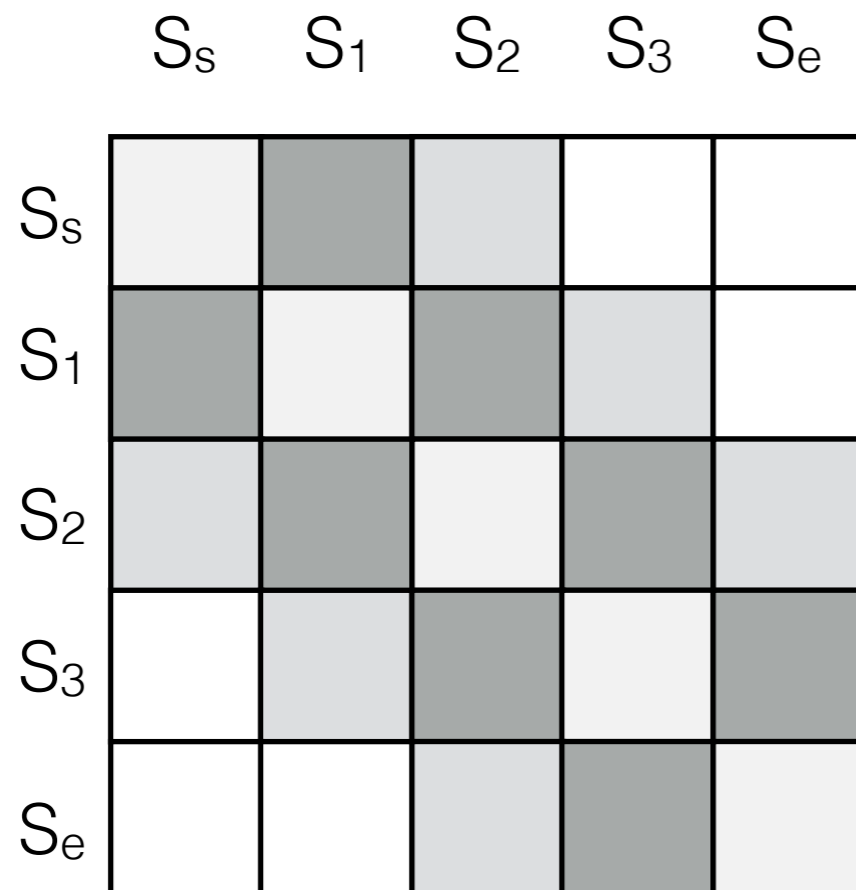
One Way





# Similarity Matrices (Predictions)

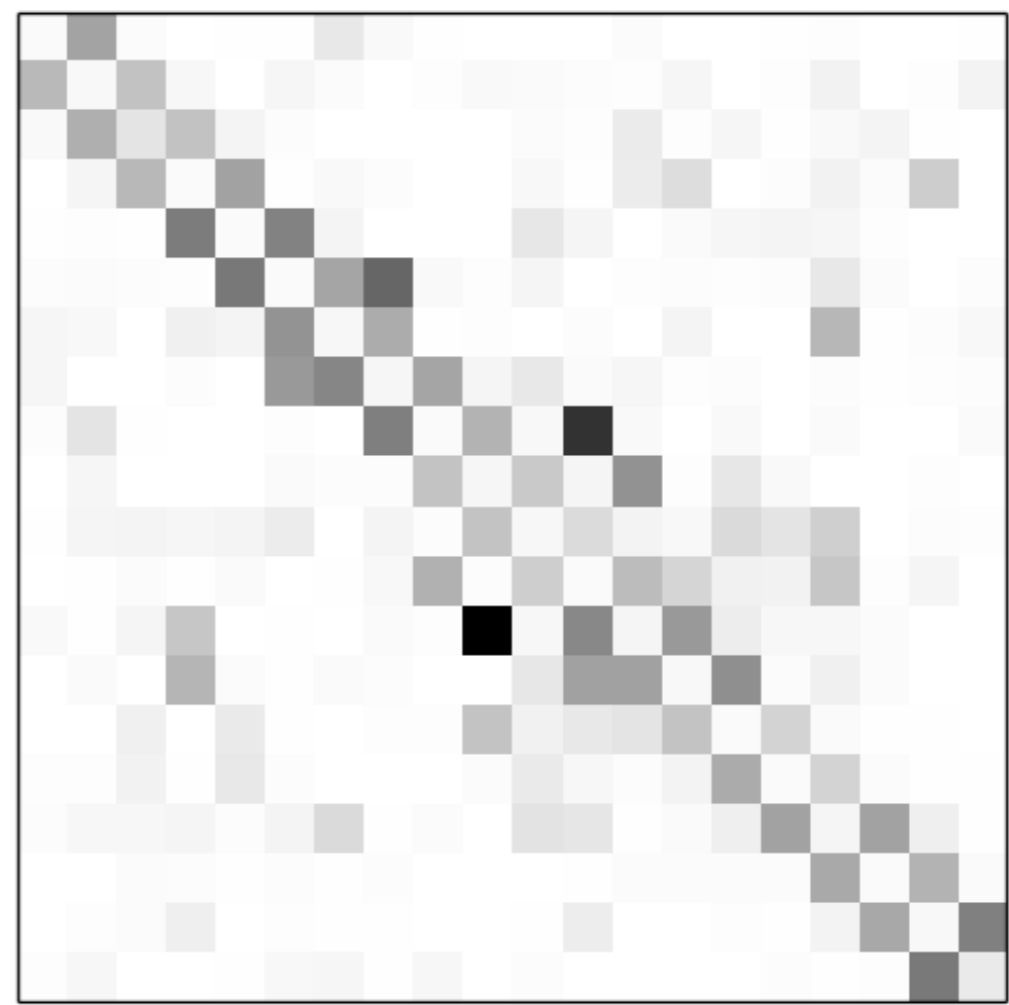
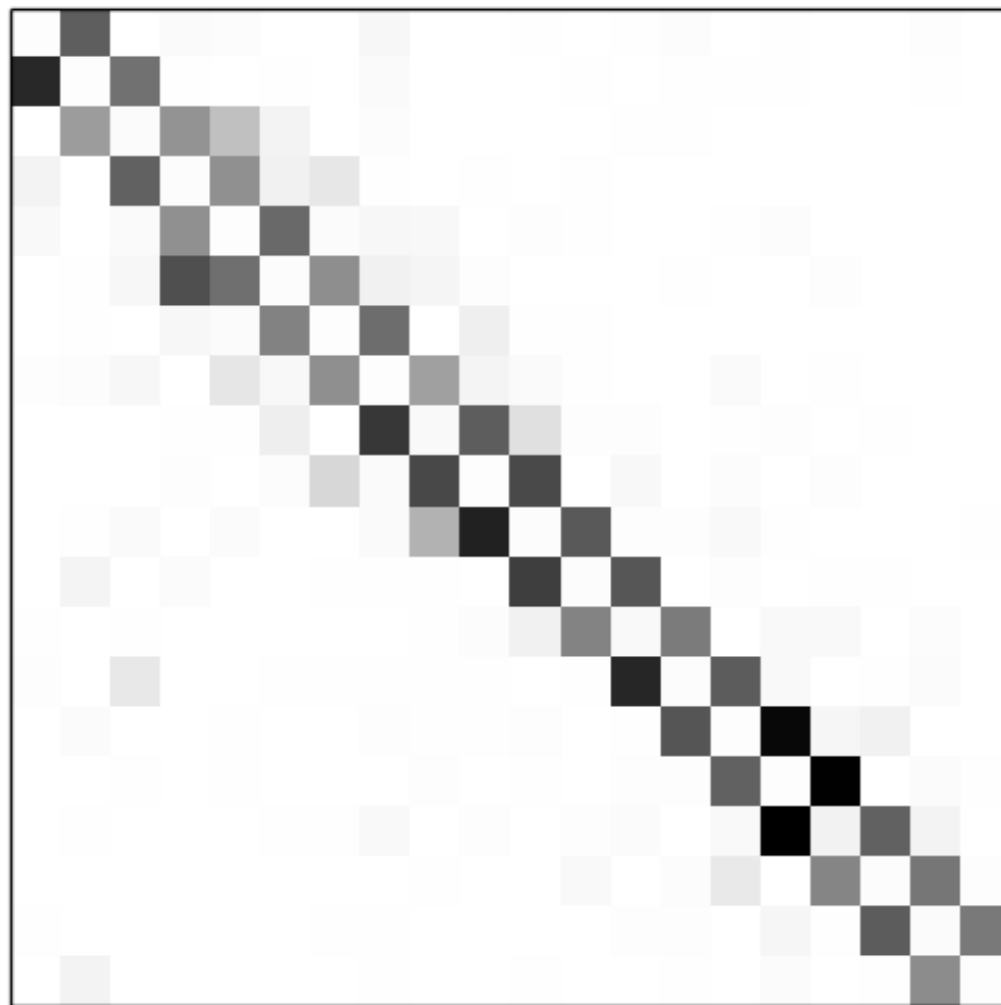
(Unbiased Random Walk Algorithm)



Greediness Parameter



# Similarity Matrices (Results)



1 Greediness Parameter 7



# Conclusions

*“Will this generate smooth playlists?” **No.** “Can we improve either algorithm?” **Yes.***

- How to Fix it?
  - Extend the data
  - Modify algorithm
- Future Work:
  - Evaluative Research

Questions?