

CS 24000 L04

Week 8 - Introduction to Linked Lists

Homework 6/7 Slides



Homework 6 - read_songs()

- Reads in data from a file in ascii format, separated by pipes
- Don't read directly into buffer!
 - Store into a temporary variable, larger than the buffer
 - Manually null-terminate strings
 - You'll find strncpy very useful (*MAX_NAME_LEN!*)
- Returns g_song_count (or appropriate errors)

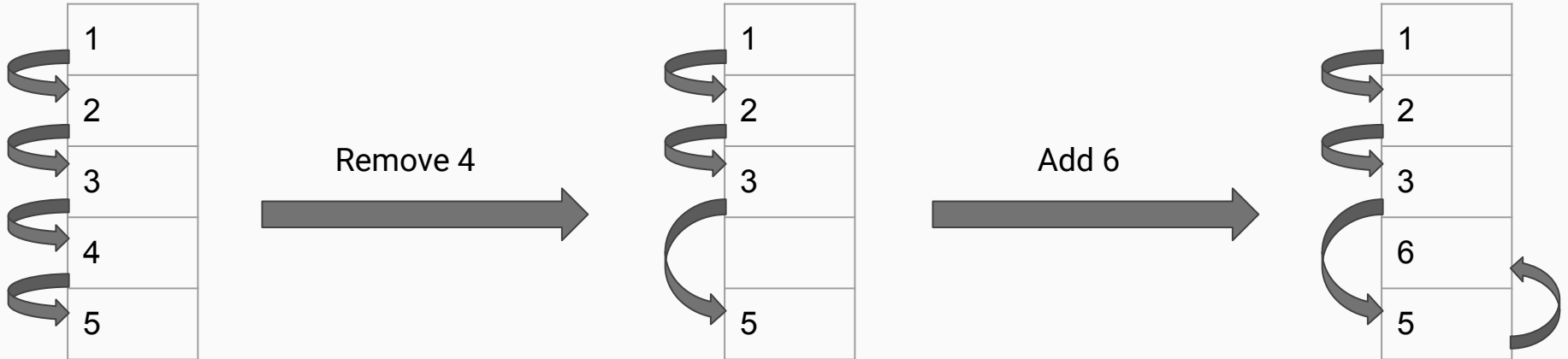
Homework 6 -

locate_matching_songs(song_name, singer_name, genre_name)

- Find the original song in g_song_array
- Compare every other song to it
 - song.song_name is a substring of song_name (*strstr!*)
 - song.singer_name is not equal to singer_name
 - song.genre_name is equal to genre_name

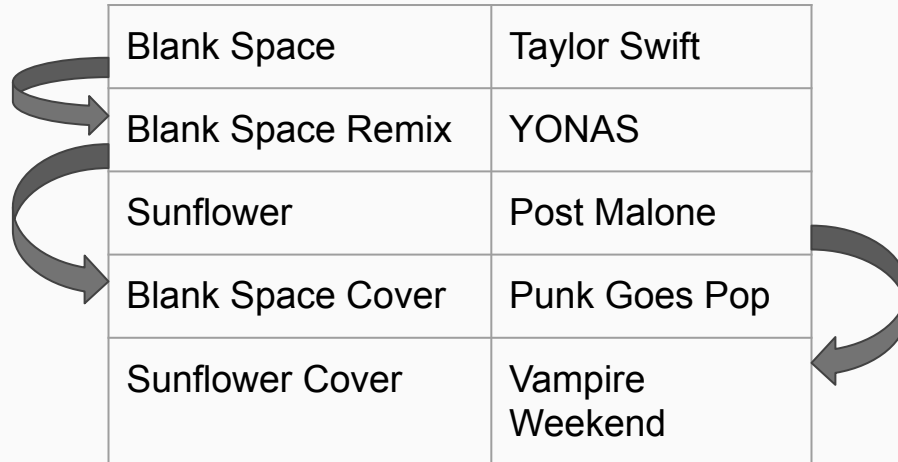
Linked Lists

- Every element in the list points to a next element in the list, or NULL
 - Initialize pointers to NULL
 - As more elements are added, set pointer to next element
 - If an element is removed, set previous next-pointer to its next-pointer



Homework 7 - Linked lists

- Similar to Homework 6, you have an array of songs, each with covers
- We want you to “link” a song to its next cover in `g_song_array`



Homework 7 - Implementation Details

- `read_songs` is almost the same as in HW6
- `link_songs` is kind of like `locate_matching_songs` from HW6
 - (You may want to reuse that function as a helper function, with some modifications)