



Date handed out: Monday 16 December 2019

Date submission due: 27 December 2019, Friday 23:55 (Cyprus Time)

Indexing Movies!

This assignment aims to help you practice binary search tree ADT, in particular AVL Tree. You will write a program that creates an index of a given list of movies.

Requirements:

In this assignment, you are given a list of movies in an external text file called "movies.txt". In this text file, you can find a number of real movies extracted from the open movie database¹.

This file includes the following information:

Title: title of the movie

Year: year of the movie

imdbID: ID of the movie for the IMBD

PosterLink: Link to visualize the poster of the movie

And sample data is as follows where each part is separated by a "," character:

Title,Year,imdbID,PosterLink

Saw,2004,t0387564,https://m.mediaamazon.com/images/M/MV5BMjE4MDYzNDE1OV5BMl5BanBnXkFtZTcwNDY2OTYwNA@@._V1_SX300.jpg

Love Actually,2003,t0314331,https://m.mediaamazon.com/images/M/MV5BMTY4NjQ5NDc0Ni5BMl5BanBnXkFtZTYwNjk5NDM3._V1_SX300.jpg

Saw IV,2007,t0890870,https://m.mediaamazon.com/images/M/MV5BNjc0NjUyNzg3MF5BMl5BanBnXkFtZTYwODMxOTM3._V1_SX300.jpg

A Man Apart,2003,t0266465,https://m.mediaamazon.com/images/M/MV5BMTk4NzEwNjk1N15BMl5BanBnXkFtZTYwNTEzNTk5._V1_SX300.jpg

Your task here is to process this file, and generate an AVL tree based on the **movie title**. If there is more than one movie with the same title, they should then be stored together in the same node. In your node structure, you need to make sure that you can accomodate a movie with more than one year of release, more than one IMDBID and also more than one PosterLink. Based on this data representation, you need to write a program that provides the following functionalities to the user:

1. **Display the full index:** This will display the full AVL tree constructed. For traversal, you need to display the movies alphabetically sorted based on their titles.
2. **Display the movies where their title contains a specific keyword:** This will display the details of the movies (Title, Year, imdbID, and PosterLink) that contains the specified keyword in

¹ <http://www.omdbapi.com/>

their title. For example if the search keyword is "Saw", then based on the above list, it should display following;

Saw,2004,t0387564,https://m.mediaamazon.com/images/M/MV5BMjE4MDYzNDE1OV5BMT5BanBnXkFtZTcwNDY2OTYwNA@@._V1_SX300.jpg

Saw IV,2007,t0890870,https://m.mediaamazon.com/images/M/MV5BNjc0NjUyNzg3MF5BMT5BanBnXkFtZTYwODMxOTM3._V1_SX300.jpg

Please note that if the movie has more than one release, they should also be displayed here.

3. **The year with the maximum number of movies:** This will display the details (Title, Year, imdbID, and PosterLink) of the movies within the year which has maximum number of movies

Programming Requirements:

You will start by taking the file name as a command line argument and then you will need to implement at least the following functions:

- **read_data:** This function will mainly process the external file. As an input, it will take the file name and it will return an AVL tree.
- **insert_movie:** This function will take an AVL tree, and the details of a movie, and then it will try to insert the movie to the AVL tree. The movie will be inserted into the tree based on the title of the movie. If the movie with the given title exist, then you will add the details of that release to the given node for that movie.
- **display_index:** This function will mainly take an AVL tree and display the data in the tree in alphabetical order according to the title of the movie.
- **most_popular_year_movies:** This function will mainly take an AVL tree and display the details of the movies within the year which has maximum number of movies. In the comment part of this function, discuss the complexity of this function based on your current representation of data. You also need to discuss if there is a way you could improve this. Please make sure that in your comments you clearly highlight your discussion on this issue.
- **display_movies_keyword:** This function will mainly take an AVL tree and will find and display the movies that contains the specified keyword in their title. In the comment part of this function, discuss the complexity of this function based on your current representation of data. You also need to discuss if there is a way you could improve this.

Please note that in this assignment, you can make use of the functions in the **string.h** library and similar external libraries. **You cannot assume about the number of movies in this external file.**

Submission Requirements:

In this assignment, you need to have a header file (avltree.h) which includes the major functionality of the AVL Tree ADT. If you will use other ADTs, you need to create a separate header file for each of them. You also need to have a C source file (indexingmovies.c) that includes the main function and other functions. You need put all these files into the "cng213a3" folder and then submit the compressed version of the folder to ODTU-CLASS. If you do not follow this structure, you will lose %10 from the overall grade.

Programming Style Tips!

Please follow the modular programming approach. In C programming, we use functions referred to modules to perform specific tasks that are determined/guided by the solution. Remember the following tips!

- Modules can be written and tested separately!
- Modules can be reused!
- Large projects can be developed in parallel by using modules!
- Modules can reduce the length of the program!
- Modules can also make your code more readable!

Sample run:

You can find the sample run of the program for the first 5 entries in the attached text file.

```
>>>Welcome to Movie Analysis at IMDB<<<
-Menu--
1. Display the full index of movies
2. Display the movies where their title contains a specific keyword
3. Display the year with maximum number of movies
4. Exit
Option:1

Movie Index
-----
A Funny Man
A Man Apart
Armitage III Dual Matrix
Blackie the Pirate
Crazy Stupid Love
Enter the Matrix
Falling Down
Happy Campers
Harlock Space Pirate
I Saw the Devil
I Saw the Sun
Love Actually
Love Me If You Dare
Saw
Saw II
Saw III
Saw IV
Saw The Final Chapter
Saw V
Saw VI
Shakespeare in Love
Snow Falling on Cedars
The Black Pirate
```

The Crimson Pirate
The Matrix
The Matrix Reloaded
The Matrix Revisited
The Matrix Revolutions
The Pirate
The Pirate Bay Away from Keyboard
The Pirate Fairy
The Pirate Movie
The Princess and the Pirate
The Texas Chain Saw Massacre
The Twilight Saga Breaking Dawn Part 1
The Twilight Saga Breaking Dawn Part 2
The Twilight Saga Eclipse
The Twilight Saga New Moon
Twilight

--Menu--

1. Display the full index of movies
2. Display the year with maximum number of movies
3. Exit

Option:2

Enter the Keyword: Saw

I Saw the Devil2010†1588170https://m.media-amazon.com/images/M/MV5BNmVkMGMOZTQ†NDQwNS00NzgwLTkwYWY†NWE3MWFiOGUwODg5XkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_SX300.jpg

I Saw the Sun2009tt1347521https://m.media-amazon.com/images/M/MV5BODU1ZWU4MGMtOGJhYS00Y2Q1LTkxZTU0NjU4YzgzZDFmMWZkXkEyXkFqcGdeQXVyMzUxOTQ2NDU@._V1_SX300.jpg

Saw2004tt0387564https://m.media-amazon.com/images/M/MV5BMjE4MDYzNDElOV5BMTU5BanBnXkFtZTcwNDY2OTYwNA@@._V1_SX300.jpg

Saw II2005tt0432348https://m.media-
amazon.com/images/M/MV5BMjY4Mjg4YTgtZWU2MC00MzVlTg3MDgtYzUyYzU1NGMyMmU5
XkEyXkFqcGdeQXVyNjU0OTQ0OTY@. V1_SX300.jpg

Saw III2006tt0489270https://m.media-
amazon.com/images/M/MV5BNWVjMzgWMTctZmZjNC00ZmEOLThiNTUyYzkyM2RkYWlZyY2Y2XkE
yXkFacGdeQXVyNiEyNDAYMzl@._V1_SX300.jpg

Saw IV2007tt0890870https://m.media-amazon.com/images/M/MV5BNjc0NjUyNzg3MF5BMT5BanBnXkFtZTYwODMxOTM3_V1_SX300.jpg

Saw The Final Chapter2010tt1477076https://m.media-amazon.com/images/M/MV5BMjl0NTEwNTgwNF5BMl5BanBnXkFtZTcwMDM5MTU5Mw@@_V1_SX300.jpg

Saw V2008tt1132626https://m.media-
amazon.com/images/M/MV5BMzc2ZWQ4YTYtZDY0Zi00YzZkLTQxNTMtMTE3MDE3NzQvNic2XkE

--Menu--

1. Display the full index of movies
2. Display the movies where their title contains a specific keyword
3. Display the year with maximum number of movies
4. Exit

Option:3

A Man Apart 2003 tt0266465 https://m.media-amazon.com/images/M/MV5BMTk4NzEwNjk1N15BMTI5BanBnXkFtZTYwNTEzNTk5._V1_SX300.jpg

Enter the Matrix 2003 tt0277828 https://m.media-amazon.com/images/M/MV5BMjA4NTYwNjk0M15BMTI5BanBnXkFtZTgwODk3MDMwMTE@._V1_SX300.jpg

Love Actually 2003 tt0314331 https://m.media-amazon.com/images/M/MV5BMTY4NjQ5NDc0Ni5BMTI5BanBnXkFtZTYwNjk5NDM3._V1_SX300.jpg

Love Me If You Dare2003 tt0364517 https://m.media-amazon.com/images/M/MV5BNjIwOGJhY2Q1MTA5Yi00MDhlLWE5OTgtYmlzZDZlM2UwZjMyXkEyXkFqcGdeQXVyNTA4NzY1MzY@._V1_SX300.jpg

The Matrix Reloaded2003 tt0234215 https://m.media-amazon.com/images/M/MV5BODE0MzZhZTgtYzkwYi00Yml5LTlhZWYtOWRmNWE5ODk0NzIxXkEyXkFqcGdeQXVyNjU0OTQ0OTY@._V1_SX300.jpg

The Matrix Revolutions2003 tt0242653 https://m.media-amazon.com/images/M/MV5BNzNiZTZjMDctZjYwNi00NzljLWlwN2Q1ZWZmYmJiYzQ0MTk2XkEyXkFqcGdeQXVyNTAyODkwOQ@@._V1_SX300.jpg

Important Rule:

If your code does not compile due to syntax errors, you will automatically get zero.

Grading:

Your program will be graded as follows:

Grading Point Mark	(out of 100)
AVL Tree Data Structure	5
Processing data file (read_data)	15
Inserting/Updating a node in the tree (insert_movie)	25
Displaying the index (display_index)	10
Displaying the movies in a year with maximum number of movies (most_popular_year_movies) (complexity discussion – 2pts)	15
Displaying the movies that contains the specified keyword (display_movies_keyword) (complexity discussion – 2pts)	15
The main function	15

NOTE: Remember to have good programming style (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.). Each of the items above will include 10% for good programming style. Good programming style also includes modularity approach explained above.