

Objective: To gain experience using a binary search tree (BST) by implementing a movie search.

To start the project: Download hw6.zip file and extract it. The hw6 directory contains the lab 10 binary search tree implementation (binary_search_tree.py and tree_node.py) which you can use.

The Assignment: Professor Bob is a movie buff, but like most professors he is absent-minded. He has a text file movieData.txt which contains the names and year(s) a movie was made.

```
...
A Tailor Made Man <1931>
A Tale of Five Cities <1951>
A Tale of Two Cities <1935> <1958> <1980> <1984>
A Talent for Murder <1984>
A Talk in the Dark <1992>
A Tall Man Executes a Jig by Irving Layton <1986>
...
```

Bob would like to be able to interactively search for all movies containing specified word(s) in the title.

The main menu should contain the following options:

- start a new search - allows the user to enter one or more words in the title to search. This should report the number of movies that match (he might not want to see 100s of movie titles)
- refine the search - allows the user to enter one or more additional words to refine the search. This should also report the number of movies that match.
- display the results of the current search: complete movie titles and the year(s) they were made
- print the results of the current search to a user-specified text file

At the start of the program, you should create an empty BST based dictionary and fill it by reading movie entries from the movieData.txt file. Each dictionary entry will have a word for its key and a list of movie entries containing that word in their titles.

On a search, use the longest word to search the dictionary for a list of movies. Use this list to find the matches if other search words were entered.

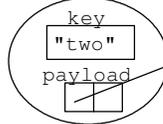
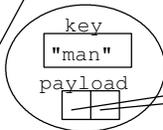
Implement AND fully test the your movie-search program. Part of your grade will be determined by how robust your program runs (i.e., does not crash) and how user-friendly/intuitive your program is to use.

Submit all necessary files (binary_search_tree.py, tree_node.py, moiveData.txt, etc.) with your movie-search program file(s) as a single zipped file (called hw6.zip) electronically at

https://www.cs.uni.edu/~schafer/submit/which_course.cgi

BST
Dictionary

Moive title strings pointed at by each word:



"A Tailor Made Man <1931>"

"A Tall Man Executes a Jig by Irving Layton <

"A Tale of Five Cities <1951>"

"A Tale of Two Cities <1935> <1958> <1980> <

