

Project: project04-castella [project04-autocomplete] C:\Users\julia\IdeaProject

Project: Alt-1

- src
- datastructures
 - ArrayDeque
 - LinkedDeque
 - TrieMap
- interfaces
 - ICollection
 - IDeque
 - IDictionary

```
11 private static Map<String, IDeque<String>> titles = new HashMap<>();
12
13 2 usages blank *
14 public static void populateTitles() {
15     Map<String, String> moviemap = ID_MAP;
16     IDeque<String> suffixdeque = new ArrayDeque<>(initial_size: 1); //initialize an IDeque to add to titles
17     for (String movie : moviemap.keySet()) { // loop through each movie given
18         movie += " "; //add a space so we can get the substring
19         for (int i = 0; i < movie.length(); i++) { // loop through the length of the movie
20             suffixdeque.addBack(movie.substring(0, i)); //add movie to the ideque in the back
21             i = movie.indexOf(" ", i); //reset i to be at the space
22         }
23         titles.put(movie, suffixdeque);
24         suffixdeque.clear();
25     }
26 }
```

Run: D Tests

Tests failed: 4, passed: 1 of 5 tests - 530 ms

Test	Time
<default package>	530 ms
HashMapAutocompleterTest	530 ms
General complete Tests	289 ms
Test complete("mission: impossible")	141 ms
Test complete("the avengers")	85 ms
Test complete("age of")	63 ms
Edge Case complete Test	66 ms
Test complete("this")	66 ms
Does not use or import disallowed classes	175 ms

Impossible - Fallout", "Mission: Impossible III", "Mission: Impossible", "Mission: Impossible - Ghost Protocol"] in any order
ossible - Fallout", "Mission: Impossible III", "Mission: Impossible", "Mission: Impossible - Ghost Protocol" in []

2 usages blank *

```
@ public static IDeque<String> complete(String term) {
    IDeque<String> movieresults = new ArrayDeque<>(initial_size: 1); //initialize the ideque we want to return
    int prefix_length = term.length();
    for (String movie : titles.keySet()) {
        IDeque<String> suffixes = titles.get(movie);
        for (String suffix : suffixes) {
            if (term.equals(suffix.substring(0, prefix_length))) {
                movieresults.add(movie);
            }
        }
    }
    return movieresults;
}
```