DSA Problems

William Schultz

September 26, 2022

Merge k Sorted Lists

- **Problem**: Given a set of k linked lists, each which are individually sorted in ascending order, merge all k lists into one sorted list.
- **Solution Idea**: The basic approach is to essentially just perform the *merge* step of merge-sort. That is, if we are given a set of already sorted lists, we can merge them all into one sorted lists by repeatedly popping the smallest element from the remaining, non-empty lists and appending it to the output list.
- **Key Concepts**:
  - ***Mergesort Merging***
  - ***Linked List Manipulation***

  The essence of the solution is very straightforward as long as you know and understand the ideas behind mergesort i.e. knowing the core idea that you can merge a set of already sorted lists by incrementally choosing the smallest element from each.

Remove duplicates from sorted linked list

- **Problem**: Given a sorted linked list, remove any duplicates from the list.
- **Solution Idea**: Iterate through the linked list, but at each node look ahead to see how many nodes in front of you contain an identical value to your own. Update your current "next" pointer to point to the first node after this block of identical nodes in front of you. Since the list is sorted, you know that any duplicates of the current value must be directly in front of you.
- **Key Concepts**:
  - ***Linked List Iteration***
  - ***Linked List Deletion***
  - ***Sorting for Duplicate Detection***

  The underlying insight in the solution is to recognize that sorting a list can be used an easy mechanism for detecting duplicates. That is, in a sorted list, all duplicates of a particular item will always appear in contiguous "blocks". Once you recognize this fact, then implementing the solution mostly requires a standard application of linked list iteration and linked list item deletion. Namely, that to delete an item n_2 from a linked list that appears in a list as $n1 \rightarrow n_2 \rightarrow n_3$, you simply update the "next" pointer of n_1 to point to n_3 instead of n_2. Recall that a basic linked list node is a $LinkedListNode(val, next)$ structure, where val is the value of that node, and $next$ is a pointer to the next item in the list.