

Programming and Data Structures with Python 2026

Assignment 1

29 Sep 2026, due 5 Oct 2026

Dictionary implementation of lists

Recall the implementation of (linked) lists using dictionaries discussed in class. A Python file with the code discussed in class is included with this assignment.

1. The function `insertlist(l,v)` inserts `v` at the beginning of the list, like `l.insert(0,v)` in Python. We want to extend this to a function `insertposlist(l,i,v)` that inserts `v` at position `i` in the list, like `l.insert(i,v)` in Python. . If `i <= 0` this function should behave like `insertlist(l,v)` and if `i >= length(l)` it should behave like `appendlist(l,v)`.
 - Write an iterative implementation `insertposlisti(l,i,v)`.
 - Write a recursive implementation `insertposlisttr(l,i,v)`.
 2. Write an iterative implementation `deletelisti(l,v)` that deletes the first occurrence of `v` in `l`, if any.
 3. Extend `deletelist(l,v)` to `deletelistall(l,v)` that removes *all* occurrences of `v` in `l`. You may implement this iteratively or recursively.
 4. Modify `deletelist(l,v)` to `deletelistlast(l,v)` that removes the *last* occurrence of `v` in `l`, if any. You may implement this iteratively or recursively.
 5. Run some experiments to determine the relative efficiency of the iterative and recursive versions of `appendlist()`, `insertposlist()` and `deletelist()`.
-

Instructions

- Submit your solution through Moodle as a single Python notebook.
 - Add documentation to explain at a high level what your code is doing.
 - For 1–4, create representative testcases and include the output on these test cases in your submission.
 - For 5, include the output of your experiments in your submission, along with your analysis of the results.
-