CSC 148: Introduction to Computer Science

Week 7

Data Structure Tradeoffs

University of Toronto Mississauga,

Department of Mathematical and Computational Sciences

Comparing Two List Implementations

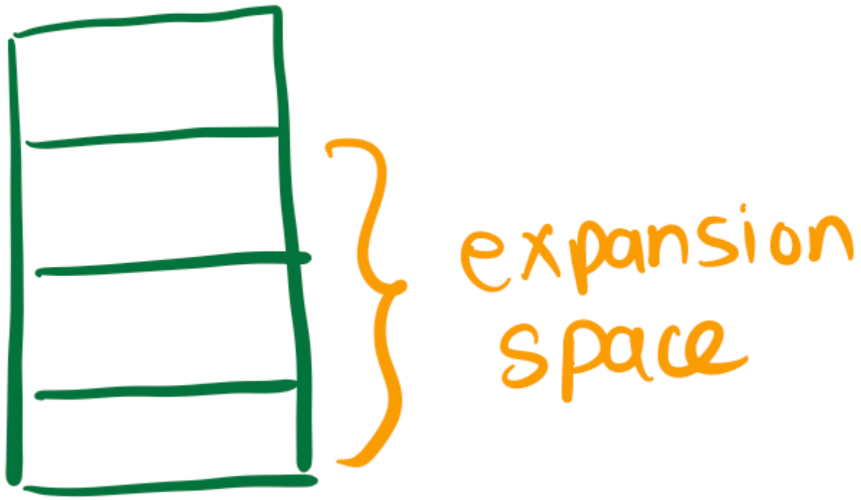
- Previously, we used a linked list to implement the List ADT … but why bother?

- Today, we'll compare Python's built in list to a new list that we build out of a Python dict.

Goal: See that choosing the *implementation* of a data structure involves analyzing tradeoffs.

How Does a Python List Work?

- A Python list uses a contiguous piece of memory.



- When a list is created, a small amount of memory is allocated.
- More memory is allocated when needed.

Superlist

We’re going to use a Python dict to implement a list. We’ll call it a Superlist, because dictionaries make everything better. (\s)

{ 0: ‘first’, 1: ‘second’, 2: ‘third’ }

- The key is the index, the value is the item at that index.

Which is Better?

How should we evaluate our Superlist versus Python’s list?

What cases should we consider?

Which is Better?

How should we evaluate our Superlist versus Python’s list?

What cases should we consider?

- Inserting at the front
- Inserting in the middle
- Inserting at the end
- And we’ll look at space complexity if time allows