

CS 350 Project 2, Parts A&B

Project 2A: Design and Implement an ADT for Doubly-Linked Lists

Project 2B: Use this ADT to Build a Mass Transit Query System

This is a two-stage project that has two due dates. For the first phase of the project, design and implement in Python an abstract data type (ADT) for doubly-linked lists. Your design and implementation should provide an interface that fully separates the implementation from any code that uses the ADT. If the implementation were to change, code using your ADT should remain fully functional without any changes. Use Python's class mechanism to implement this ADT.

At this point you should be familiar with linked lists. We will discuss doubly-linked lists in class (or "will have discussed" depending on when you are reading this). You have some latitude in what you choose to include or how you actually implement your design, but you should keep the second phase of this project in mind so that you have everything you'll need. Operations you'll want to include:

- Creation of doubly-linked lists

- Adding, removing, and changing items on lists

- Displaying lists (essential for debugging)

- Searching (up or down), retrieving, and displaying information on lists

- Initializing lists, i.e., building a list, from a file or other data structure

You should feel free to add to these.

Most of these operations should be very simple to write. Remember that you will be using this ADT in the second phase of the project. Design your ADT so it can be used as a module.

In the second phase of the project you will use the doubly-linked list to implement a query system for a mass-transit system. You are free to use the transit system and city of your choice, but you should take care in making an appropriate selection. For example, the Atlanta Marta is too simple while the NY subway or London tube are much too complex. You do not need to model the entire system (feel free to omit stops) and may make reasonable simplifications such as eliminating branches (per our discussion in class). You'll also want to stick to linear systems. For example, the Spartanburg Sparta bus system (or any bus system) has loops that you won't be able to cleanly fit into a linked list. The Boston MTA or DC Metro might be reasonable choices. Be sure to turn-in a map of the transit system you are modeling.

With your system, the user should be able to

Query the system to see what line a stop is on

Print a list of all the stations on a line in order

Print a list of all stations on a line in a given direction, e.g., north of Haymarket

Provide directions for traveling between points on the same line including a list of intervening stops

*Provide directions for traveling between points on different lines including a transfer point

Your code should work from the data representations you use—do not hardcode information into the system. The system should be capable of working with a different data set. For example, if you model the DC Metro, I should be able to change to the Boston MTA by changing the map information and nothing else. In addition to the doubly-linked list class and ADT definitions, you should define ADTs and classes as appropriate for the remainder of your code.

You are *required* to work in pairs on this project, preferably with someone you haven't worked with before. Turn in one printed copy of your code and one complete demonstration of your code per team. Turn in an accurate map of the transit system you are modeling that reflects what you have modeled. You should also send me a copy of the source code as an email attachment. Each individual student should send me a brief email telling how long they worked on the project and describing any problems they encountered. The team project will be given one grade and both members will receive that grade.

Part A & Transit Map Due: Friday, October 17 by the start of class.

Part B Due: Monday, October 20 by the start of class.

These due dates are negotiable.