Homework 1

Due Friday, February 1, at 5:00 PM

"I'm not a regular coder. I'm a COOL coder."
-Regina George, 2004

Handing In

To hand in a homework, go to the directory where your work is saved and run `cs0160_handin hwX` where X is the number of the homework. Make sure that your written work is saved as a .pdf file, and any Python problems are completed in the same directory or a subdirectory. You can re-handin any work by running the handin script again. We'll only grade your most recent submission. To install stencil Python files for a homework, run `cs0160_install hwX`. Please leave room between questions and in the margins on your pdf so that your grader can leave feedback on your work. **You will lose points if you do not hand in your written work as a .pdf file.**

Overview

The following applies to all homeworks:

- We will not accept paper handins. You must submit all written work as a PDF. For information on how to make PDFs of your work, see the very handy **PDF guide**. Please make sure to **NOT** include any identifying information, e.g. name or banner info on your handin, as we are anonymizing grading - we appreciate your help with this.

- Please follow proper pseudocode formatting guidelines. See our **pseudocode standards**. If you use LaTeX, use the `newalg` or `algorithmic` packages or our CS16 `pseudo` environment to format your pseudocode. For even more wizardry, check out the Docs section of our website for a LaTeX handout and tips on improving your pseudocode.

1 Creating a Homework Directory

Since there are no Python files and no install script for this homework, you'll need to create a homework directory for yourself! If you do not already have a cs0160 directory, then you can make one by typing the following command in your terminal:

```
mkdir ~/course/cs0160
```

To make a directory specifically for hw1, type the following commands in your terminal:

```
mkdir ∼/course/cs0160/hw1
cd ∼/course/cs0160/hw1
```

Run the handin script from the hw1 directory once you have added your PDF. Please note that if you are using LaTeX, creating a PDF will generate other files. Only hand in the .pdf file. (When you hand in, you may see that something called './' is also handed in. This just means you are turning in the assignment directory and is fine).

2 Written Problems

Problem 1.1

CS 16 Collaboration Policy

If you have not already, take a few minutes to read the **CS16 Collaboration Policy** and the **CS16 Hours Policy**. Once you have finished, fill out **this Google form** regarding the collaboration policy. Next, type "I have read and agree fully to abide by the CS16 Collaboration Policy and the CS16 Hours Policy" and sign the statement digitally by typing your full name. NB: We realize this is in conflict with the note in the Overview to not include identifying information on your handin. This problem will be the only time in the semester when we request, or allow, you to include it. ***Do not forget to include your signed statement with your homework!***

Problem 1.2

Check out course website + Piazza!

This is really important! Important announcements and all assignments will be posted on the course website, so please make sure to check it out! Also, you must log in to **Piazza** and post on the **welcome thread** by the time that this homework is due. Please introduce yourself, including your favorite 2000's icon! In addition, in the same post please include a question to either a TA or the staff as a whole. This can be as specific as to 'Where is the PDF that explains SSH?' to as general as 'What is your favorite color?' **Be sure to include your CS login somewhere in the post.**

Note that Piazza collects data of your activities and shares it with companies. You can opt out of sharing your data by following the steps in **this link**.

Problem 1.3

Regina wants to give Cady a makeover, including a brand new wardrobe. Regina knows she has plenty of extra clothing, so she plans on giving what she doesn't need to Cady. She has her mom look through her closet and make a list of all the clothing she has. Your job is to help Regina go through the list and determine which article of clothing appears consecutively the most.

Write pseudocode for a function, `findConsecutiveMax(array)`:

findConsecutiveMax: [array of characters] → character

Purpose: Given an array of characters, returns the character that occurs the greatest number of *consecutive* times

Examples:

- `findConsecutiveMax([c,c,g,g,h,a,a,a,f,f,a,b,c,c,b,c])` should return `'a'`, because it appears three times in a row– more than any of the other characters. If multiple characters appear consecutively the maximum number of times, your function should return the character whose sequence comes first in the array.

- `findConsecutiveMax([c,a,a,a,b,c,c,c])` returns `'a'`.

- `findConsecutiveMax([s,h,a,k,e,w,e,i,g,h,t])` returns `'s'`.

- If the array is `Empty` or `Null`, return an empty string (`""`).

Problem 1.4

Cady's parents are out of town, so she has planned to hold a ”small get-together” at her house. Unfortunately, the get-together ends up being anything but small. Cady wants to make sure she greets everyone in the order they come in and does not miss anyone (in particular, Aaron), but she only has two stacks. How could you implement the methods of a queue using only those two stacks to help Cady?

Remember, stacks work in a Last In First Out (LIFO) order, meaning the last element that was pushed will be the first element to be popped.

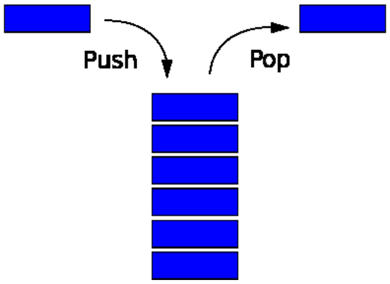


Figure 1: A stack

Conversely, a queue works in First In First Out (FIFO) order, meaning the first element enqueued will be the first element to be dequeued.

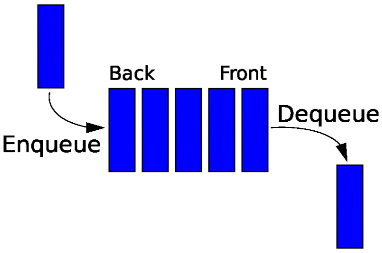


Figure 2: A queue

If you need a refresher on this material, take a look at the lecture slides from CS15 on stacks and queues.

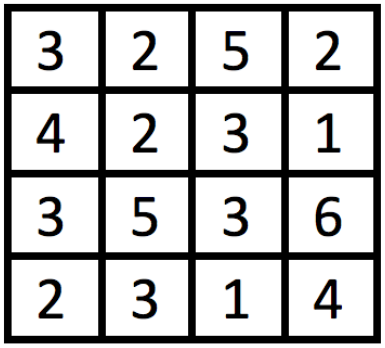
Your task: Write pseudocode for the following methods of a queue: `enqueue(value)` and `dequeue()`. The `dequeue()` method should return the element being removed from the queue. Assume you have instance variables $s1$ and $s2$ that represent the two stacks, which already have the methods `push(value)`, `pop()` (which returns the top element), and `isEmpty()` (which returns a boolean). You may call these methods freely on both stacks, and can assume that they will function as per the standard stack definition described above. You can assume that the `pop()` method handles returning an element in the case that the stacks are empty (i.e. calling `pop()` on an empty stack will return null). **You may not use any other data structures to store your data** (like `arrays` or `ArrayLists`).

Problem 1.5

Regina, Cady, Karen, and Gretchen have a photo of Principal Duvall to add to the Burn Book, but it's too large to fit on the page. Your task is to help them find the seam of least-total-importance to shrink the photo to the appropriate size while maintaining Principal Duvall's proportional features.

Below is a table of calculated importance values. Your task is to find the optimal seam from the top of the table to the bottom. You should provide a sequence of column-indices, each differing from the last by at most 1, with corresponding pixels being the least-total-importance vertical seam. For example, a non-optimal seam [0,1,2,3] would refer to the seam going through col 0 → 3, col 1 → 2, col 2 → 3, col 3 → 4. Hand-simulating this problem will be very helpful in understanding how finding a seam works.

Show your work by turning in the 4x4 cost and directions tables (see slides for a refresher) in addition to the column-indices representing the optimal seam. For the directions table, use 1 to indicate down and to the right, 0 for directly below, and -1 for down and to the left.



3	2	5	2
4	2	3	1
3	5	3	6
2	3	1	4