



# Polytechnic Tutoring Center

## Exam 2 Review - CS 1134, Fall 2020

**Disclaimer: This mock exam is only for practice. It was made by tutors in the Polytechnic Tutoring Center and is not representative of the actual exam given by the CS Department.**

1. Given an unmodified basic implementation of a stack, singly/doubly linked list, queue, dequeue what are the runtimes for the following:

Find the minimum value in a stack of ints

Insert at front of a singly linked list

Push onto a stack

Remove the last element from a singly linked list

Insert into middle of a doubly linked list

2. Assume that the function takes in the front node of the list, a sample node class is defined below. (assume that the first node has valid data and is not an empty header node)

Write a function that takes a singly linked list and recursively prints it out in reverse.

```
class Node(object):
def __init__(self, data=None, next_node=None):

    self.data = data

    self.next_node = next_node
```

Code:

3. Circle the faster runtime:

$O(\log(\log(n)))$  or  $O(n\log(n))$

$O(n\log n)$  or  $O(n^{1.25})$

$O(2^n)$  or  $O(n!)$

4. Given a string with an undefined number of open or closed parentheses and braces: ( [ and ]),

determine if the parentheses are balanced “()[([])” is balanced.

“[(())]” is NOT balanced.

You may assume you have predefined implementations of an array, stack, queue, and dequeue. You may assume the string passed as a parameter will only consist of ‘(, ’), ‘[, ’] characters.

Code:

5. Given a dequeue of characters, write a function that determines if the dequeue currently holds a palindrome. (you may modify the contents of the deque). Assume that a predefined implementation of a dequeue has been provided with the following methods:

- `deq.push_front(element):`
- `deq.push_back(element):`
- `deq.pop_front()`
- `deq.pop_back()`
- `deq.front()`
- `deq.back()`
- `deq.__len__()`

inserts an element to the front of the dequeue  
inserts an element to the back of the dequeue  
removes the front of the dequeue  
removes the back of the dequeue

returns the front element from the dequeue  
returns the back element from the dequeue  
returns the number of elements in the dequeue

A palindrome is a string that is identical if read from front to back or from back to front. For example: "racecar" is a palindrome

Code:

6. Given a node in a singly linked list, write a function to remove all subsequent instances of a single number passed as a parameter. Assume the list has at least one element

```
class Node(object):  
    def __init__(self, data=None, next_node=None):  
  
self.data = data  
self.next_node = next_node
```

Code:

7. Write a function that passes in two sorted doubly linked lists, `srt_lst1` and `srt_lst2`, ordered smallest to largest, and recursively merges them into one sorted doubly linked list, ordered smallest to largest.