Little 1a: Data Structures and Recursion

Submitted by:

Collaboration: You are encouraged to collaborate with up to 5 other students, but all work submitted must be Whiteboard only and write-ups must be composed completely independently. List the names of all of your collaborators. Do not seek published solutions for any assignments. If you use any published resources (other than those provided by the course) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff.

Problem 1: Insert and Remove

Complete the table below (the first row is done for you).

Data structure	Method to insert	Worst case time to insert	Method to remove	Worst case time to remove
Queue	`enqueue(x)`	$O(1)$	`dequeue()`	$O(1)$
Stack	`push(x)`		`pop()`	
Max Heap	`insert(x)`		`extract_max()`	
Hash Table	`insert(x)`		`remove(y)`	

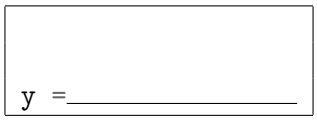
Problem 2: Draw the data structure

Each of the data structures below can be implemented as an array. Fill in the boxes so that the values could represent the data structure's contents after executing the provided psuedo-code. Also give the value of the variable `y`.

1. **Queue**

```
my_list = [0, 57, 35, 99, 101, 57, 102, -5, 0, 99, -8]
my_queue = new Queue()
for item in my_list:
    my_queue.enqueue(item)
y = my_queue.dequeue()
```

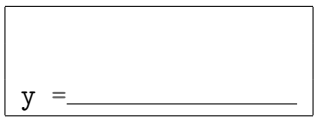
Contents of `my_queue`:



2. **Stack**

```
my_list = [0, 57, 35, 99, 101, 57, 102, -5, 0, 99, -8]
my_stack = new Stack()
for item in my_list:
    my_stack.push(item)
y = my_stack.pop()
```

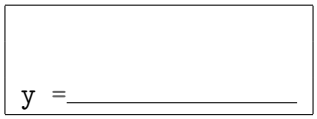
Contents of `my_stack`:



3. **Max Heap**

```
my_list = [0, 57, 35, 99, 101, 57, 102, -5, 0, 99, -8]
my_heap = new Max_Heap()
for item in my_list:
    my_heap.insert(item)
y = my_heap.extract_max()
```

Contents of `my_heap`:



Problem 3: Trees

1. How many elements will a complete binary tree of height n contain?

2. What is the minimum height of a binary tree containing 107 elements?

Problem 4: Describe the function

What does the following recursive function do (explain in 1 sentence)? We will say `my_list.pop(i)` removes the element at index `i` from `my_list`.

```
def recursion(my_list):
    if my_list.length() < 2:
        return True
    return (my_list[0] < my_list[1]) and recursion(my_list.pop(0))
```

Source Declarations

I Collaborated with the following people when completing this assignment:

I Consulted the following resources when completing this assignment: