### CSC 300 Spring 2019 Final Exam preparation

### Practice Problems

* + Demonstrate the Θ-complexity of the functions below. Recall that in order to demonstrate that f(n) = Θ(g(n)) you must find 3 positive integers C1, C2 and x such that

C1\*g(n) ≥ f(n) ≥ C2\*g(n) for all n ≥ x

* + 1. f(n) = 2n2 - 3
    2. f(n) = .5n + 2log(n+1)

**ANSWERS: Theta(n^2), Theta(n)**

* + Give the Θ-complexity of each of the loop constructions below

(a)

int sum = 0;

for (int i=0; i<n; i++) {

int j = i;

while (j > 0) {

sum += j;

j /= 2;

}

}

**ANSWER: Theta(n log n)**

(b)

int sum = 0;

int i=0;

for (int i=0; i<n/2; i++)

sum++;

for (int j=i; i<n; i++)

sum++;

**ANSWER: Theta(n), because they are sequential loops**

(c)

int sum = 0;

int i=0; j=n\*n;

while (i++ < j--)

sum++;

* + Show the results of enqueueing the following items into an initially empty priority queue: (2 A), (6 B), (5 C), (18 D), (41 E), (1 F), (16 G), (24 H), (9 I). Assume the priority queue is implemented as a heap/complete binary tree stored in an array. The length of the array is 9. Lower priority items come first’

**ANSWER:** (priorities only)

[2]

[2, 6]

[2, 6, 5]

[2, 6, 5, 18]

[2, 6, 5, 18, 41]

[1, 6, 2, 18, 41, 5]

[1, 6, 2, 18, 41, 5, 16]

[1, 6, 2, 18, 41, 5, 16, 24]

[1, 6, 2, 9, 41, 5, 16, 24, 18]

Then show the queue after the first 3 items (with the lowest priorities) are dequeued.

[2. 6. 5, 9, 42, 18 , 10, 24]

[5. 6. 10, 9, 4, 2, 18, 24]

[6, 9, 10, 24, 42, 18]

* + What comparisons are performed in a binary search when searching for the number 2 is called on an array containing the numbers 0-14 in ascending order?

**ANSWER:** 7 – 3 – 1 - 2

* + Explain which of the data structures that we have discussed during this quarter would be most useful in each of the following applications. You may choose from the following data structures: stack, queue, list, or priority queue.
    - 1. You are building a system which processes orders that have been placed in an on-line store. The orders should be processed in the order that they were made. -- **queue**
      2. You are writing a text editor, and wish to implement an "undo" feature, which behaves that way that the Control-z keystroke does in Microsoft Word. -- **stack**
      3. You are building a system that maintains a wait-list of passengers on a flight who are waiting for an upgrade to first class. The passenger with the most frequent flier miles is the first to be upgraded when a first class seat becomes available – **priority queue**.
      4. You are building a system that will be used to maintain a list of people who are employed by a company. The system should maintain the list in alphabetical order by the employees' last names, and should support operations to add and remove employees from the list as they are hired by or leave the company. -- **stack**

### Practice Problems, part 2

* + Assume that a list has been implemented using an array and the “wrap-around” technique discussed during lecture. **Redraw the list as a linked list.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| “3” | “0” | “0” |  | “C” | “S” | “C” |

front=4, back=2

**ANSWER:**

**LinkedList300 Node300 Node300 Node399 Node300 Node300 Node300**

**First item “C” item “S” item “C” item “3” item ‘0’ item ‘0’**

**Last next mext next next next next**

null

* + What does this LinkedList method do? I’ve included parts of the LinkedList300 class that are relevant to the answer.

**public** **class** LinkedList300<T> **implements** List300<T> {

**private** **class** Node<T> {

**public** T item;

**public** Node next;

**public** Node(T item, Node next) {

**this**.item = item;

**this**.next = next;

}

}

**private** Node<T> first;

**private** Node<T> last;

**private** **int** size;

**public** **void** mystery() {

**if** (size == 0) **return**;

size--;

**if** (first.next == last)

last = first;

first = first.next;

}