Sample Review Questions & Answers for 91.404 Material

I : Function Order of Growth (20 points)

1. (5 points) Given the following list of 3 functions:

$\sqrt{3}\,(n^2)$ $25 + (8\sqrt{n})$ $(n \lg n) - 6$

Circle the one ordering below in which the 3 functions appear in <u>increasing</u> asymptotic growth order. That is, find the ordering $\mathbf{f_1}$, $\mathbf{f_2}$, $\mathbf{f_3}$, such that $\mathbf{f_1 = O(f_2)}$ and $\mathbf{f_2 = O(f_3)}$.

(a) $\sqrt{3}\,(n^2)$ $25 + (8\sqrt{n})$ $(n \lg n) - 6$

(b) $25 + (8\sqrt{n})$ $(n \lg n) - 6$ $\sqrt{3}\,(n^2)$

(c) $(n \lg n) - 6$ $\sqrt{3}\,(n^2)$ $25 + (8\sqrt{n})$

2. (5 points) Given the following list of 3 functions:

$(1/6)n!$ $(2^{\lg n}) + 5$ $9(\lg(\lg n))$

Circle the one ordering below in which the 3 functions appear in <u>increasing</u> asymptotic growth order. That is, find the ordering $\mathbf{f_1}$, $\mathbf{f_2}$, $\mathbf{f_3}$, such that $\mathbf{f_1 = O(f_2)}$ and $\mathbf{f_2 = O(f_3)}$.

(a) $(1/6)n!$ $(2^{\lg n}) + 5$ $9(\lg(\lg n))$

(b) $(2^{\lg n}) + 5$ $9(\lg(\lg n))$ $(1/6)n!$

(c) $9(\lg(\lg n))$ $(2^{\lg n}) + 5$ $(1/6)n!$

Sample Review Questions & Answers for 91.404 Material

For problems 3 and 4, assume that:

$f_1 = \Omega(n \lg n)$ $f_2 = O(n^2)$ $f_3 = \Theta(n)$ $f_4 = \Omega(1)$

For each statement: Circle TRUE if the statement is true.
Circle FALSE if the statement is false.
Circle <u>only one</u> choice.

3. (5 points) $f_1 = \Omega(f_3)$ **TRUE** **FALSE**

4. (5 points) $f_3 = O(f_2)$ **TRUE** **FALSE**

Sample Review Questions & Answers for 91.404 Material

II: Solving a Recurrence (10 points)

In each of the 3 problems below, solve the recurrence by finding a closed-form function **f(n)** that represents a tight bound on the asymptotic running time of **T(n)**.

That is, find **f(n)** such that $T(n) = \Theta(f(n))$.

1. **(5 points)** Solve the recurrence : $T(n) = T(n/4) + n/2$

[You may assume that $T(1) = 1$ and that **n** is a power of 2.]

Circle the one answer that gives a correct closed-form solution for T(n)

(a) $\Theta(n)$ **(b)** $\Theta(n^2)$ **(c)** $\Theta(n \lg n)$

2. **(5 points)** Solve the recurrence : $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$

[You may assume that $T(2) = 1$ and that **n** is of the form 2^{2^k}.]

Circle the one answer that gives a correct closed-form solution for T(n)

(a) $\Theta(n^2 \lg n)$ **(b)** $\Theta(n \lg^2 n)$ **(c)** $\Theta(n \lg(\lg n))$

Sample Review Questions & Answers for 91.404 Material

III: PseudoCode Analysis (30 points)

Here you'll use the pseudocode below for two functions **Mystery1** and **Mystery3**. **Mystery1** has three arguments: **A** : an array of integers; **p, r** : indices into **A**

```
Mystery1(A, p, r)
    if p is equal to r
        then return 0
    q ← ⌊(p+r)/2⌋
    g1 ← Mystery1(A, p, q)
    print "Mystery1 result1= ", g1
    print contents of A[p]..A[q]
    g2 ← Mystery1(A, q+1, r)
    print "Mystery1 result2= ", g2
    print contents of A[q+1]..A[r]
    g3 ← Mystery3(A, p, q, r)
    print "Mystery3 result= ", g3
    print contents of A[p]..A[r]
    return g3
```

```
Mystery3(A, p, q, r)
    initialize integer array B to be empty
    bb ← p
    pp ← p
    qq ← q+1
    while pp <= q and qq <= r
      do if A[pp] <= A[qq]
           then B[bb] ← A[pp]
                pp ← pp + 1
                bb ← bb + 1
           else B[bb] ← A[qq]
                qq ← qq + 1
                bb ← bb + 1
    while qq <= r
      do B[bb] ← A[qq]
         qq ← qq + 1
         bb ← bb + 1
    while pp <= q
      do B[bb] ← A[pp]
         pp ← pp + 1
         bb ← bb + 1
    a ← 0
    for i ← p to r
       do A[i] ← B[i]
          if i>p
             then d ← |A[i] - A[i-1]|
          if d > a
             then a ← d
    return a
```

Sample Review Questions & Answers for 91.404 Material

(a) (10 points) For **A = <**5,2,16,9,25**>**, what output is generated by the call **Mystery1(A, 1, 5)**?

(b) (5 points) Describe in one or two sentences the mathematical meaning of the result of a call **Mystery1(A, 1, length[A])** on an arbitrary integer array **A**.

Sample Review Questions & Answers for 91.404 Material

(c) (10 points) Find a function **f(n)** that can be used to describe a tight bound on the worst-case asymptotic running time **T(n)** of **Mystery1**, where **n** is the number of elements in array **A**.

That is, find **f(n)** such that **T(n) =** Θ **(f(n))**.

(*You may ignore floors and ceilings in your analysis*.)

(d) (5 points) Does the time bound from (c) also describe the best-case and average-case asymptotic running time **T(n)** of **Mystery1**? Briefly explain.

Sample Review Questions & Answers for 91.404 Material

IV: Patriotic Trees (40 points)

This problem is based on the definition of a **PatrioticTree** in the Final Exam Handout (see attachment).

1. **(10 points)** Consider a **PatrioticTree TreeNode t** and the two **TreeNode**s **t.left** and **t.right** representing its two children. How many different colorings of the 3 **TreeNode**s **t**, **t.left** and **t.right** are possible? (circle the one correct answer below)

27 **13** **23** **9** **17**

Sample Review Questions & Answers for 91.404 Material_

2. **(5 points)** An algorithm **MAX_R_COLORING** accepts as input a list of initialized **TreeNode**s representing leaves of a **PatrioticTree** (initialization provides weight and color for each leaf). Assume that the sum of the RedWeights of the leaves exceeds the sum of the BlueWeights of the leaves. Assume also that **MAX_R_COLORING** always returns a **PatrioticTree** (built from the leaves up) in which the number of **R**-colored internal nodes is <u>maximized</u>. Then the number of **R**-colored internal nodes in the tree built by **MAX_R_COLORING** is <u>always</u>: (circle the one correct answer below)

2 f **2f - 1** **f + 1** **f - 1**

Sample Review Questions & Answers for 91.404 Material

3. **(25 points)** Given a collection of **f** leaf nodes, develop the algorithm **MAX_R_COLORING** as described in Problem 2.
In addition to all assumptions listed in the statement of Problem 2, you may assume that:

- There are **w** white leaves and they are stored in an array **W** of **TreeNode**s
- There are **r** red leaves and they are stored in an array **R** of TreeNodes
- There are **b** blue leaves and they are stored in an array **B** of TreeNodes

a) (15 points) Provide pseudo-code for the algorithm **MAX_R_COLORING**

b) (5 points) Justify the correctness of the algorithm

c) (5 points) Establish an upper bound on the algorithm's worst-case running time in terms of the total number of nodes n.

Sample Review Questions & Answers for 91.404 Material

FINAL EXAM HANDOUT

40% of the points on the Final Exam will be based on the representation described in this handout.

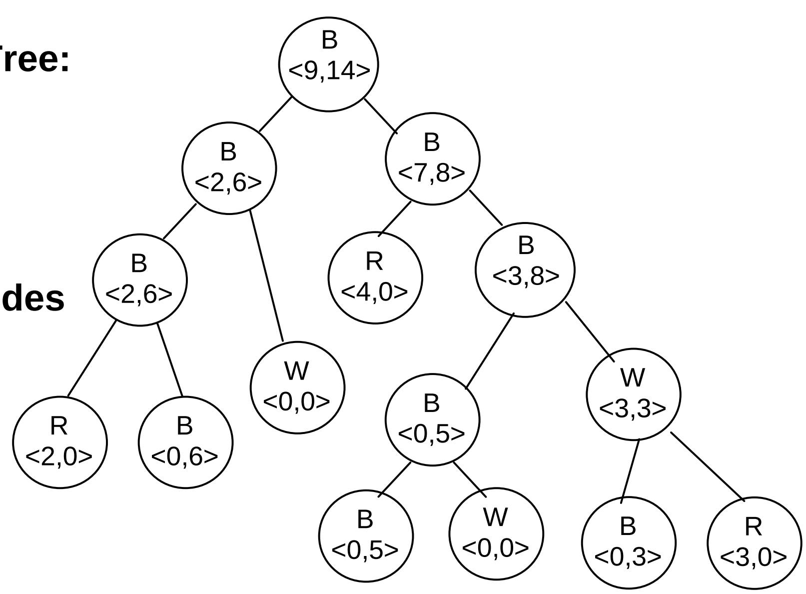
We define a new type of tree as follows:
A **PatrioticTree** is a *binary tree* of **n** nodes in which:

- Every internal node has both a left and a right child.
- Each node is labeled with a single color (RED, WHITE or BLUE, which we abbreviate as R, W, or B).
- Each node has a weight. The weight is a pair of integer values and is of the form <r,b>. The first integer value in the pair is the Red Weight of the node = Red Weight of the node's left child + Red Weight of the node's right child. The second integer value in the pair is the Blue Weight of the node = Blue Weight of the node's left child + Blue Weight of the node's right child. The color of a node is determined as follows:
  - R if its Red Weight exceeds its Blue Weight
  - B if its Blue Weight exceeds its Red Weight
  - W if its Blue Weight equals its Red Weight
- The weight of each W leaf is <0,0>.
- The weight of each R leaf is of the form <r,0>.
- The weight of each B leaf is of the form <0,b>.

EXAMPLE of a PatrioticTree:

In this example:

- **there are 8 leaf nodes**
- **there are 7 internal nodes (including the root)**
- **the total number of nodes is 15**



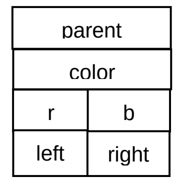
Sample Review Questions & Answers for 91.404 Material

FINAL EXAM HANDOUT
(continued)

For algorithmic purposes, we represent a PatrioticTree as follows:

A **TreeNode** represents a node of a PatrioticTree. It contains the attributes:

- **color**: a character representing its color: “R”, “B” or “W”
- **r**: an integer representing its Red Weight
- **b**: an integer representing its Blue Weight
- **parent**: a pointer to the parent of this node
- **left**: a pointer to the left subtree of this node
- **right**: a pointer to the right subtree of this node



You can access a node’s attributes using the “**.**” notation (e.g. for a **TreeNode t**, **t.color** gives its color and **t.left** accesses its left subtree).

Sample Review Questions & Answers for 91.404 Material

1: (20 points) Given the recurrence:

$$T(n) = T(n/2) + 3T(n/3) + 4T(n/4) + 7n^2$$

(a) (10 points) Show that $T(n) = O(n^3)$

(b) (10 points) Show that $T(n) = \Omega(n^{3/2})$

2: True/False (30 points) Circle TRUE if the statement is true. Circle FALSE if the statement is false. Circle only one choice.

(a) (15 points) If $f_1 = n\,2^n$, $f_2 = 2^{n+1}$ then $f_1 = O(f_2)$

TRUE **FALSE**

(b) (15 points) Clustering for dynamically resizable hash tables

Define a dynamically resizable hash table to be one in which, whenever the hash table becomes full, the size of the hash table is doubled, and all entries are removed from the small hash table and inserted into the new, larger hash table.

Consider a hash table H_1 of size **m** and a hashing function $h_1(k) = k \bmod m$. Let H_2 be of size **2m**. Suppose that open addressing with linear probing is used for both H_1 and H_2. Let the hashing function for H_2 be $h_2(k) = k \bmod 2m$.

If, for the keys $k_1, k_2, ..., k_b$ it is the case that
$k_1 \bmod m = k_2 \bmod m = ... = k_b \bmod m$
and when these keys are inserted (in the order given) into H_1 they occupy consecutive slots in H_1, then if only $k_1, k_2, ..., k_b$ are inserted into H_2 (in the same order) they also occupy consecutive slots in H_2.

TRUE **FALSE**

Sample Review Questions & Answers for 91.404 Material

3: (50 points) This is based on the Exam Handout.
distributed on 11/17 *(see attachment).*

(a) (5 points) Here we examine the amount of storage required for this representation. If each **Key Node** counts as one unit of storage and each **Neighbor Ring Node** also counts as one unit, give (and briefly justify) a tight (upper and lower) bound on the worst-case number of storage units needed for this representation as a function of **n**, the total number of **Key Node**s.

(b) (5 points) A neighbor triple is a triple of **Key Node**s $\langle \mathbf{k_1}, \mathbf{k_2}, \mathbf{k_3} \rangle$ such that each pair within the triple has the neighbor relation. (In the example in the exam handout, the nodes with keys **a**, **b**, and **c** are in a neighbor triple.) In the worst case, how many neighbor triples can there be (as a function of **n**, the total number of **Key Node**s)?

<u>Clarification</u>: We consider all neighbor triples containing $\mathbf{k_1}$, $\mathbf{k_2}$, and $\mathbf{k_3}$ to be the same.

Sample Review Questions & Answers for 91.404 Material

(c) (40 points) Using the neighbor triple definition in (b):

I. Given a reference (pointer) to a **Key Node k**, give pseudo-code to decide if **k** is part of a neighbor triple. If **k** is part of a neighbor triple, the pseudo-code should print out “yes” plus the 3 key values of a neighbor triple. (You need not find all triples; you may stop after finding the first one.) If **k** is not part of any neighbor triple, the pseudo-code should print out “no”.

II. Justify the correctness of your algorithm.

III. Give (and justify) a tight (upper and lower) bound on your algorithm’s worst-case asymptotic running time as a function of **n**, the total number of **Key Node**s.

Sample Review Questions & Answers for 91.404 Material

Exam Handout

This reinforces material from Chapters 1-12. *Date Distributed: Friday, 11/17*

The open-book exam on Monday, 11/20 will contain a 50-point question based on the scenario described below.

In this problem, there are two types of list nodes:

1) a **Key Node** that contains the attributes:

- **key**: a key value
- **info**: satellite data
- **ring**: a pointer to a linked list of **Neighbor Ring Nodes**

  We call this list a **Neighbor Ring.**

key
info
ring

2) a **Neighbor Ring Node** that contains the attributes:

- **keyNode**: a pointer to the Key Node for this Neighbor Ring
- **neighbor**: a pointer to a **Neighbor Ring Node** for a neighbor of this **Key Node**'s Neighbor Ring
- **neighborCCW**: a pointer to the next **Neighbor Ring Node** in this **Key Node**'s **Neighbor Ring** (where next is in the counter-clockwise direction)
- **neighborCW**: a pointer to the next **Neighbor Ring Node** in this **Key Node**'s **Neighbor Ring** (where next is in the clockwise direction)

neighborCCW
neighborCW
neighbor
keyNode

You can access a node's attributes using the "**.**" notation (e.g. for a **Key Node k**, **k.key** gives its key and for a **Neighbor Ring Node n**, **n.neighbor** gives its neighbor).

The neighbor relation is symmetric: if **a** is a neighbor of **b**, then **b** is a neighbor of **a**.

<u>You may assume that there is a total of **n** Key Nodes.</u>

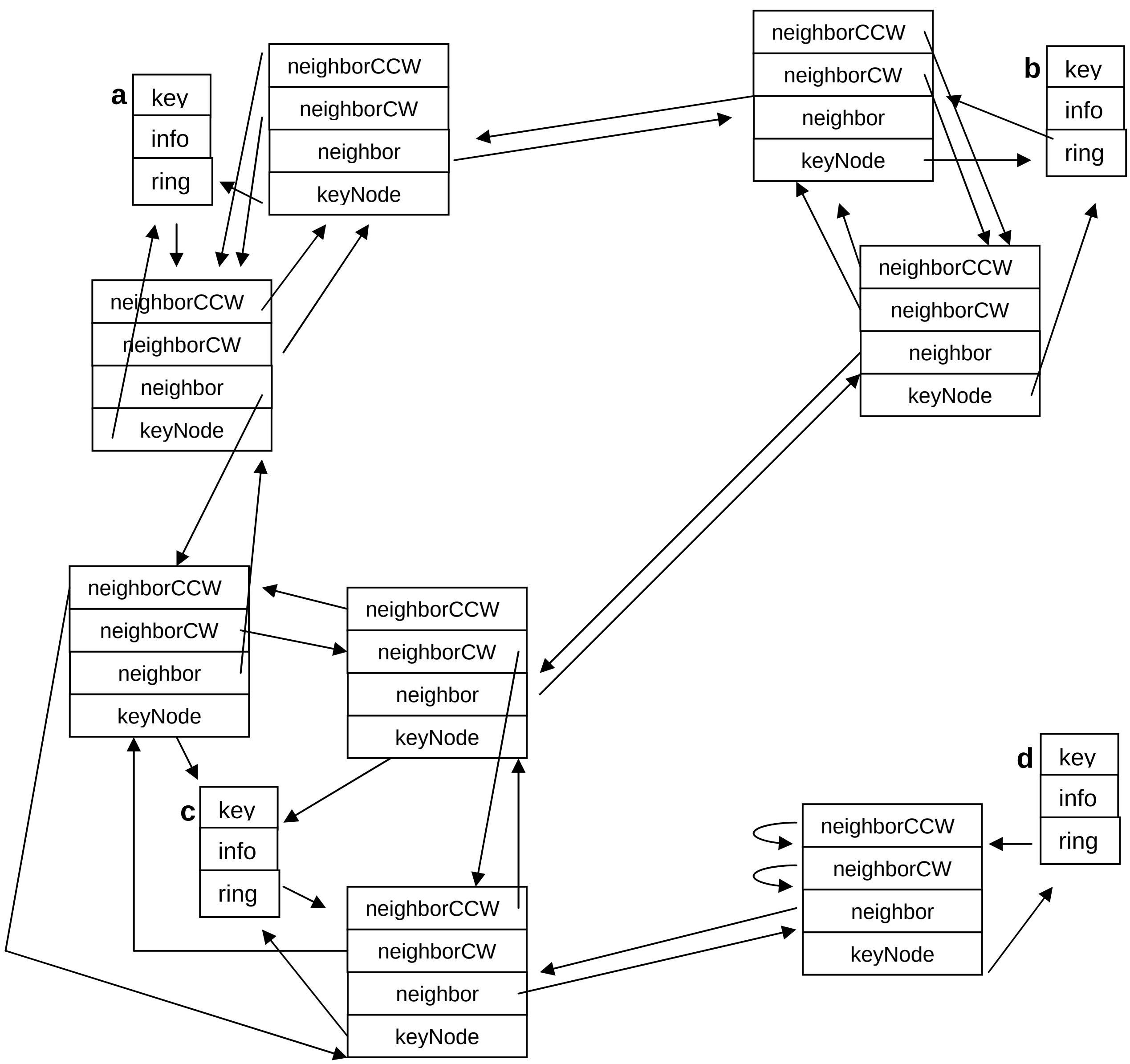
(See Example on Next Page)

[Note: Representations that use neighbor rings are often useful in graphics and geometric modeling applications.]

Sample Review Questions & Answers for 91.404 Material

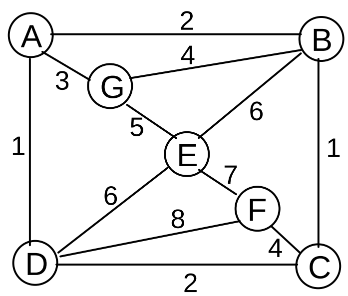
Example:

- Nodes with key values **a** and **b** are neighbors
- Nodes with key values **a** and **c** are neighbors
- Nodes with key values **b** and **c** are neighbors
- Nodes with key values **c** and **d** are neighbors



Sample Review Questions & Answers for 91.404 Material

3. (10 points) *Minimum Spanning Trees*: For the undirected, weighted graph below:

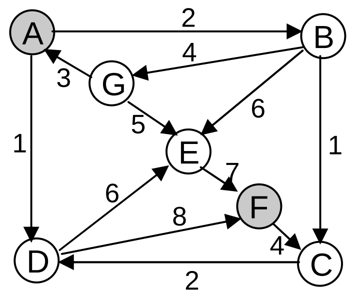


Show 2 different Minimum Spanning Trees. Draw each using one of the 2 graph copies below. Thicken an edge to make it part of a spanning tree.

What is the sum of the edge weights for each of your Minimum Spanning Trees?

Sample Review Questions & Answers for 91.404 Material

4. (5 points) For the directed, weighted graph below, find the shortest path that begins at vertex A and ends at vertex F.



a) (4 points) List the vertices in the order that they appear on that path

b) (1 point) What is the sum of the edge weights of that path?

Sample Review Questions & Answers for 91.404 Material

I : Multiple Choice (10 points)

1. (5 points) Function Order of Growth

Given the following list of 3 functions:

$$(n^2 \lg n) - 6 \qquad (n \lg^3 n) + 12n \qquad (5n/6)^2$$

Circle the one ordering below in which the 3 functions appear in increasing asymptotic growth order. That is, find the ordering $\mathbf{f_1}$, $\mathbf{f_2}$, $\mathbf{f_3}$, such that $\mathbf{f_1 = O(f_2)}$ and $\mathbf{f_2 = O(f_3)}$.

(a)	$(n \lg^3 n) + 12n$	$(5n/6)^2$	$(n^2 \lg n) - 6$
(b)	$(5n/6)^2$	$(n \lg^3 n) + 12n$	$(n^2 \lg n) - 6$
(c)	$(5n/6)^2$	$(n^2 \lg n) - 6$	$(n \lg^3 n) + 12n$

2. (5 points) Solve a Recurrence

In the problem below, solve the recurrence by finding a closed-form function **f(n)** that represents a tight bound on the asymptotic running time of **T(n)**.

That is, find **f(n)** such that $\mathbf{T(n) = \Theta(f(n))}$.

Solve the recurrence: $\mathbf{T(n) = 9T(n/3) + (n+2)(n-2)}$

[You may assume that $\mathbf{T(1) = 1}$ and that $\mathbf{n}$ is a power of 3.]

Circle the one answer that gives a correct closed-form solution for T(n)

(a) $\Theta(n)$ **(b)** $\Theta(n^2)$ **(c)** $\Theta(n \lg n)$ **(d)** $\Theta(n^2 \lg n)$

II: True/False (15 points)

For each problem in this section: Circle TRUE if the statement is true.
Circle FALSE if the statement is false.
Circle only one choice.

1. (5 points) Function Order of Growth: What can you conclude?

Assume that:

Sample Review Questions & Answers for 91.404 Material

$f_1(n)= O(2^n)$ $f_3(n)= \Theta(f_2(n))$ $f_3(n)= \Omega(n)$ $f_4(n)= \Omega(n^3)$

If we also know that $f_3(n) = \Omega(f_4(n))$, then we can always conclude that $f_2(n) = \Omega(n^3)$.

TRUE FALSE

2. (5 points) Bounds on Alg vs Problem/ Types of input

Given a problem P, suppose that:

- **someone designs algorithm A_1 to solve P**
- **someone designs algorithm A_2 to solve P**
- **$T(A_i)$ represents the asymptotic running time of algorithm A_i**
- **someone constructs a proof that solving an arbitrary instance of P requires at least time in $\Omega(n)$.**
- **the designer of algorithm A_1 claims that algorithm A_1 is in $O(2^n)$ for worst-case inputs**
- **the designer of algorithm A_2 claims that algorithm A_2 is in $O(n^2)$ for worst-case inputs**

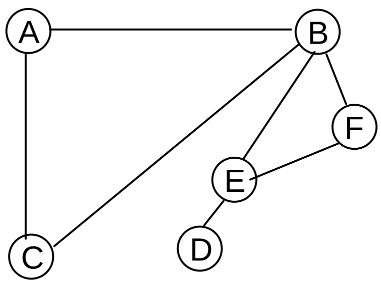
If all these proofs and claims are correct, then we know that

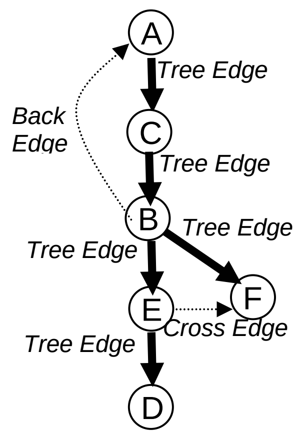
$T(A_2)$ is in $\Omega(n)$ and $T(A_2)$ is in $O(n^3)$

TRUE FALSE

3. (5 points) The tree shown below on the right can be a DFS tree for some adjacency list representation of the graph shown below on the left.

TRUE **FALSE**





1: True/False (20 points = 10 points each)

For each statement: Circle TRUE if the statement is true.
Circle FALSE if the statement is false.
Circle <u>only one</u> choice.

(a) if $\mathbf{f_1 = \Theta(f_2)}$ then $\mathbf{f_1} = O(\mathbf{f_2})$ and $\mathbf{f_2} = O(\mathbf{f_1})$ **TRUE** **FALSE**

(b) if $\mathbf{f_1} = O\,(\mathbf{n^2})$ and $\mathbf{f_2 = \Omega(n)}$ and $\mathbf{f_3 = n\ lgn}$ **TRUE** **FALSE**

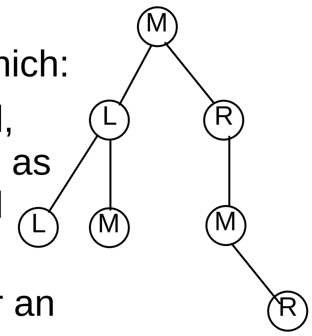
then $\mathbf{f_3} = O\,(\mathbf{f_1})$ and $\mathbf{f_3 = \Omega(f_2)}$

and $\mathbf{f_3 = \Omega\,(n)}$ *implies* $\mathbf{f_3 = \Omega\,(f_2)}$

Sample Review Questions & Answers for 91.404 Material_

2: Multiple Choice (30 points)

For each statement: circle the one answer that is true.



We define a new type of tree as follows: An **i/3-tree** is one in which:

- Each node may have children of the following types: left-child, middle-child, or right-child. Each node is labeled with its type as follows: L for left-child, M for middle-child and R for right-child (we consider the root to be a middle child).
- Each node has **at most i** children (see picture on the right for an example of a **2/3-tree**).

A **complete i/3-tree** is an i/3-tree in which each node has exactly i children (*clarification*: *at all levels except the lowest*).

Solution to extra credit:

5 points **(a)** The number of ***different*** **complete 3/3-trees** with 13 nodes is:

(i) 27 (ii) 1 (iii) 3 (iv) 18 (v) 9

10 points **(b)** The number of ***different*** **complete 1/3-trees** with 3 nodes is:

(i) 27 (ii) 16 (iii) 3 (iv) 18 (v) 9

Sample Review Questions & Answers for 91.404 Material

15 points **(c)** If we construct a **1/3-tree *randomly*** so that, for each node, the probability that it has a left child is the same as the probability that it has a right child and the same as the probability that it has a middle child, then the **expected** number of **middle children** (including the root node) in a complete 1/3-tree with 3 total nodes is:

(i) 9 (ii) 18/16 (iii) 3 (iv) 15/9 (v) 2

Extra Credit: 20 points

The number of ***different* complete 2/3-trees** with 7 nodes is:

(i) 27 (ii) 16 (iii) 3 (iv) 18 (v) 9

Derive a general formula for the number of different complete i/3-trees, where the tree has j levels.

Sample Review Questions & Answers for 91.404 Material_

Answer:

Name:______________________________

1: Function Order of Growth (20 points)

Put the 3 functions below into increasing asymptotic growth order:

That is, find **g_1(n)**, **g_2(n)**, **g_3(n)** so that:

g_1(n) is in O(g_2(n)) and **g_2(n) is in O(g_3(n))**

Justify your ordering by giving values of **c** and **n_0**.

$16^{\lg(n^2)}$	$n^7 \lg n$	$n^2 \lg(\lg n) - 8\lg^6 n$

g_1(n) = **g_2(n) =** **g_3(n) =**

Show that: **g_1(n) is in O(g_2(n))**

Show that: **g_2(n) is in O(g_3(n))**

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:______________________________

2: Function Order of Growth (10 points)

Given: $f_1(n)$ is in $\Omega(\lg n)$ $f_2(n)$ is in $\Theta(\lg(\lg n))$ $f_3(n)$ is in $O(\lg^2 n)$

If we also know that $f_3(n)$ is in $\Omega(f_1(n))$,
can we conclude from these 4 statements that $f_1(n)$ is in $O(\lg^2 n)$?

Why or why not?

(page 3 of 9)

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:________________________

3: Solving Recurrences (30 points)

In **(a)** and **(b)** below, find a closed-form solution for **T(n)**.

That is, find **f(n)** such that **T(n) is in** Θ **(f(n))**.

You may assume that $n = 9^k$ and that $T(1) = 1$.

(a) (10 points) $T(n) = 3T(n/9) + 6n$

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:______________________________

(b) (20 points) $T(n) = 3T(n/9) + 9\sqrt{n}\,\log_9 n$

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:____________________________

4: PseudoCode Analysis (40 points)

Mystery1 has 3 arguments:

- **A** : a one-dimensional array of **n** integers (assume that $n = 2^k$)
- **p**: an integer value representing an index for array **A**
- **r**: an integer value representing an index for array **A**

```
Mystery1(A, p, r)
    if p < r
        then  q ← ⌊(p+r)/2⌋
              Mystery1(A, p, q)
              Mystery1(A, q+1, r)
              Mystery2(A, p, r)
```

```
Mystery2(A, p, r)
    for j ← p+1 to r
        do key ← A[ j ]
           i ← j-1
           while i >= p and A[ i ] > key
               do  A[ i+1 ] ← A[ i ]
                   i ← i-1
          A[ i+1] ← key
```

(page 6 of 9)

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:____________________________

(a) (10 points)

Show the final result of executing the call **Mystery1(A, 1, 8)**

on the 8-element array **A = <13, 24, 7, 8, 5, 25, 38, 99>**

(b) (10 points) Give an example of a worst-case input for **Mystery1**

(page 7 of 9)

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:____________________________

(c) (10 points) Describe the worst-case asymptotic running time of

Mystery1 by developing a recurrence of the form:

$$T(n) = aT(n/b) + \Theta(f(n))$$

(d) (10 points) Solve the recurrence you developed in (c).

That is, find **g(n)** such that $T(n) = \Theta(g(n))$.

(page 8 of 9)

Sample Review Questions & Answers for 91.404 Material

UMass Lowell CS *91.404 (Section 201)* *Fall, 2001*

Name:______________________________

EXTRA CREDIT: (10 points)

Given the recurrence below, find a closed-form solution for **T(n)**.
That is, find **f(n)** such that **T(n) is in** Θ **(f(n))**.

You may assume that **$n = 9^k$** and that **T(1) = 1**.

$$T(n) = 3T(n/9) + 18n^{1/3}$$

(page 9 of 9)

Sample Review Questions & Answers for 91.404 Material

3. (50 points) This question is based on the Exam Handout distributed on 10/27 *(see attachment to exam).*

Clarifications:

- *The delivery schedule* **S_i** *only needs to contain the messageID, not the entire message.*
- *Message processing algorithm operates on entire queue to produce Delivery Schedule; no new messages added during this processing.*

For a single node v_i:

(a) (5 points) Where in the message processing algorithm is sorting required in order to produce S_i ? Explain.

(b) (30 points) Assuming that fast asymptotic running time is the most important judgment criterion, what type of sorting algorithm(s) is (are) appropriate for use within the message processing algorithm?

Justify your answer. For each sorting algorithm that you select, your answer should describe the algorithm by **writing pseudo-code** and stating:

(i) whether it is a comparison-based sort or a non-comparison-based sort

(ii) in more detail what type of sorting algorithm it is. For example, is it a MergeSort? an InsertionSort? a HeapSort? a BucketSort? a RadixSort? A variation on one of these types?

(iii) the worst-case asymptotic running time

(iv) the worst-case size of the delivery list (*clarification: this is* S_i)

(v) the amount of extra storage you use (for data structures other than C_i, Q_i, S_i)

Note: You need not write out pseudo-code that appears in our textbook; simply cite the page reference. However, if you modify pseudo-code that appears in our textbook, you must write pseudo-code for your modifications. You may assume the existence of a queue data structure that has the following O(1) time operations: enQueue(Q, x), deQueue(Q,x), (*clarification: and deQueue(Q))* isEmpty(Q). You may also assume that the operations dynamically allocate storage as needed.

(c) (15 points) Describe how your results for (b) change if we change our assumptions so that a message priority may have any integer value larger than 0.

Sample Review Questions & Answers for 91.404 Material

Exam Handout

This reinforces material from Chapters 1-11. *Date Distributed: Friday, 10/27*

The open-book exam on Wednesday, 11/1 will contain a 50-point question based on the scenario described below.

Suppose that you are working for a company that designs message processing algorithms for managing message deliveries on a network. For the purposes of the algorithm, the network is modeled as a collection of vertices in an undirected graph **G** (refer to Chapter 5 if necessary for graph definitions). Each vertex represents a node in the network. If there is a direct network connection between two nodes in the network, then the vertices representing those two nodes have an edge in the graph **G**. (Note: We assume that two nodes have a direct network connection if and only if a message sent from one node to the other does not need to pass through any additional nodes in the network.) Each node can only send messages to nodes to which it is directly connected. Each message consists of a header and a body. The header consists of metadata that describes the message: a message ID, its destination list and its delivery priority. A destination is another node in the network to which the message must be sent; we represent as an integer (e.g. 0 for vertex $\mathbf{v_0}$). A destination list is an ordered list of destinations. The message must visit its destinations in the order in which they appear in its destination list: i.e. visit the first, then the second, and, ultimately, the last. The message's delivery priority is an integer in between 1 and 10, where 10 is the highest priority value. The message ID is a unique integer for each message.

Each node/vertex $\mathbf{v_i}$ has:

- a message processor $\mathbf{P_i}$ that can execute algorithms;
- a FIFO message queue $\mathbf{Q_i}$ containing $\mathbf{m_i}$ messages;
- a list $\mathbf{C_i}$ of the nodes to which it is directly connected.

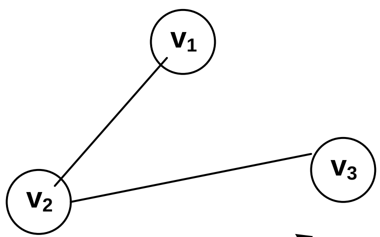
When a message arrives at $\mathbf{v_i}$, the first destination in its destination list (which is $\mathbf{v_i}$) is removed from the message header. If $\mathbf{v_i}$ is the final destination (that is, the destination list is now empty), the message is not added to $\mathbf{Q_i}$. If the next destination (which is now first in the destination list) is not in $\mathbf{C_i}$, the message is not added to $\mathbf{Q_i}$. Otherwise, the message is added to $\mathbf{Q_i}$.

Sample Review Questions & Answers for 91.404 Material

The message processing algorithm that executes on $\mathbf{P_i}$ must examine $\mathbf{Q_i}$ and produce a delivery schedule $\mathbf{S_i}$. The schedule $\mathbf{S_i}$ lists the messages that $\mathbf{v_i}$ will send. In $\mathbf{S_i}$, the messages appear in the order in which they will be sent. This order is determined based on the delivery priority and the order in which the messages appear in the message queue $\mathbf{Q_i}$. The order is consistent with the delivery priorities and the arrival order in $\mathbf{Q_i}$ is used to break ties.

Example:

G



$\mathbf{C_1}$

2

$\mathbf{C_2}$

1	3

$\mathbf{C_3}$

2

These will remain the same as algorithm(s) execute

These will change as algorithm(s) execute.
Here we show an example for one time period.

$\mathbf{v_1}$

$\mathbf{Q_1}$

messageID: 1	messageID: 2
destinationList: 2, 3	destinationList: 2
priority: 5	priority: 10

$\mathbf{S_1}$

messageID: 2	messageID: 1

$\mathbf{v_2}$

$\mathbf{Q_2}$

messageID: 4	messageID: 5	messageID: 6
destinationList: 1	destinationList: 3	destinationList: 1
priority: 6	priority: 8	priority: 8

$\mathbf{S_2}$

messageID: 5	messageID: 6	messageID: 4

$\mathbf{v_3}$

$\mathbf{Q_3}$

messageID: 3
destinationList: 2
priority: 1

$\mathbf{S_3}$

messageID: 3