15110 PRINCIPLES OF COMPUTING – EXAM 2A – FALL 2012

1	25
2	30
3	20
4	25
TOTAL	100

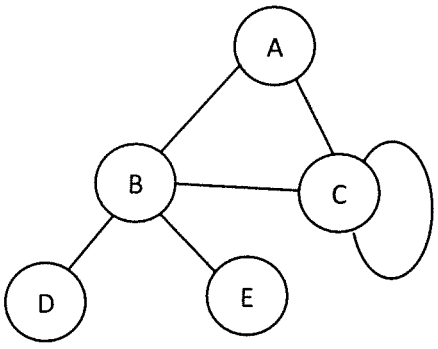
Name Answer Key Section A – G

Directions: Answer each question neatly in the space provided.

Please read each question carefully. You have 50 minutes for this exam. No electronic devices allowed. Good luck!

1. Data structures (25 points)

(1a) Recall that graphs can be represented using an adjacency matrix, which in Ruby is an array of arrays. Write the adjacency matrix for the following undirected graph to the right of the diagram.



	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	1
C	1	1	1	0	0
D	0	1	0	0	0
E	0	1	0	0	0

(1b) Given a matrix, we may observe its properties by inspecting its entries. For example, an identity matrix would always have the property such that all the numbers are 1 along the diagonal and 0 elsewhere. Given this, the adjacency matrix of an undirected graph has what interesting property?

Symmetry along the diagonal

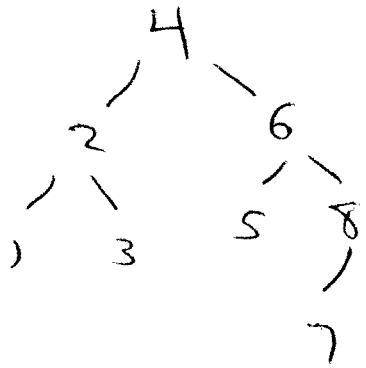
(1c) What is the maximum possible number of links in a directed graph of N nodes? N^2

(1d) The maximum degree of a graph is the largest number of links of any node. Given an adjacency matrix representation of an unweighted graph, describe an algorithm to calculate the maximum degree of the graph by examining this matrix. You don't have to write executable code, just explain in English

the steps necessary to solve the problem.
Count the number of ones in each row and return the max

(1e) A complete binary tree with N nodes has depth log N, where depth is defined to be the longest path from the root to any leaf node.

(1f) Draw a binary search tree where the nodes contain numeric keys from the set 1, 2, 3, 4, 5, 6, 7, and 8. Note that a binary tree must satisfy a specific property for it to be a binary search tree.



2. Data representation (30 points)

(2a) It's estimated that there are roughly 250,000 distinct English words. Suppose that you are given a list of these words, and you want to be able to check very quickly whether a given string is a word in English, so that you can check thousands of strings per second. Which data structure would you use for this?

Hash table

(2b) Given a true array, where the items are layed out in sequential memory locations (as opposed to a Ruby array, which is actually a linked list), what is the cost of accessing an item at a given index? Write your answer using the big O notation.

O(1)

(2c) Consider converting the 5 bit binary number 10011 (base 2) to decimal form (base 10). Fill in the blanks in the given equation where the expression on the right hand side represents the decimal value.

$10011_{[2]} = (1 \times 2^{4}) + (0 \times 2^{3}) + (0 \times 2^{2}) + (1 \times 2^{1}) + (1 \times 2^{0}) =$ 19

16 2 1

(2d) How many distinct values can be represented using a 4-bit binary number?

16

(2e) Suppose that 4 bits are used to represent signed integers using 2's complement representation. Write the binary representation of the number -3 according to this representation. Show your work. Hint: Note that adding a positive number x to a negative number $-x$ should result in a binary number with 0000 in the rightmost 4 bits.

3 = 0011

complement = 1100

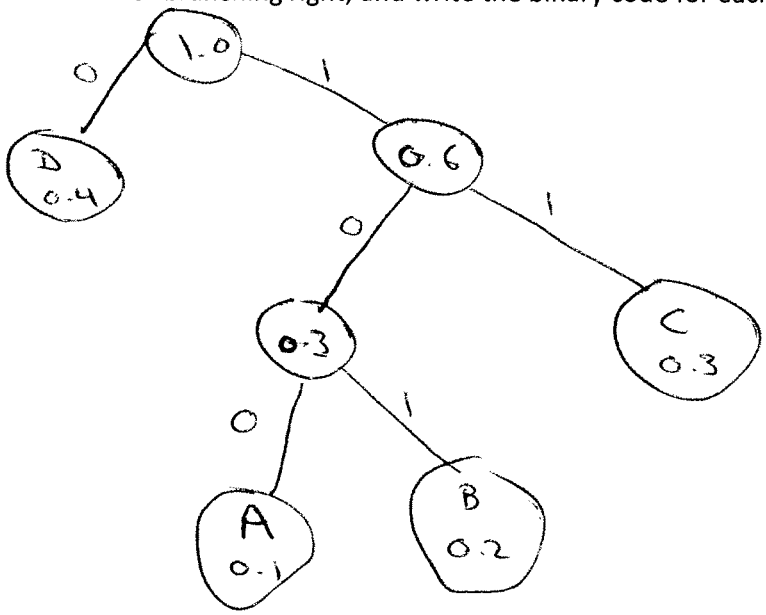
add 1 = 1101 = −3

(2f) How many bits are needed to store an uncompressed Red-Green-Blue (RGB) image from a 4 megapixel digital camera? Recall that "mega" stands for 2^{20} and RGB encoding uses 8 bits to represent the intensity of each color. Show your calculation.

4×2^{20} pixels × 3 bytes/pixel × 8 bits/byte =

$2^{22} \times 3 \times 2^{3} = 3 \times 2^{25}$ bits

(2g) Suppose that we are compressing some text consisting of the letters A, B, C, and D with frequencies 10%, 20%, 30%, and 40%, respectively. Construct a Huffman tree for this text using 0 for branching left and 1 for branching right, and write the binary code for each letter.



A = 100

B = 101

C = 11

D = 0

Other codes are possible but the lengths must match the above solution.

(2h) Suppose that we have a piece of 10-letter text consisting of letters A,B,C, D with the frequencies given in part (g), which we encode using normal the ASCII fixed length code of 7 bits per letter. What is the compression ratio obtained by Huffman encoding? Answer: 19 divided by 70. Show your work.

	Length	Freq	Expected length
A	3	0.1	0.3
B	3	0.2	0.6
C	2	0.3	0.6
D	1	0.4	0.4

1.9 bits per letter

ASCII: 10 letters × 7 bits/letter = 70 bits

Huffman: 10 letters × 1.9 bits/letter = 19 bits

3. (20 points) Suppose we want to write a recursive function `print_right_nested(x)` to print every non-nil element of a right-nested list, e.g., `[3, [1, [4, [1, [5, [9, nil]]]]]]`. First read (3a) – (3g) before writing any answers.

```
def print_right_nested(x)
    if (i) x.nil?, then
     (ii) return nil
    else
      (iii) puts x[0]
      (iv) print-right-nested(x[1])
    end
  end
```

Assume that in our initial call, x is `[3, [1, [4, [1, [5, [9, nil]]]]]]`.

(3a) What is x.length? 2

(3b) What is x[0]? 3

(3c) What is x[1]? [1, [4, [1, [5, [9, n]]]]]

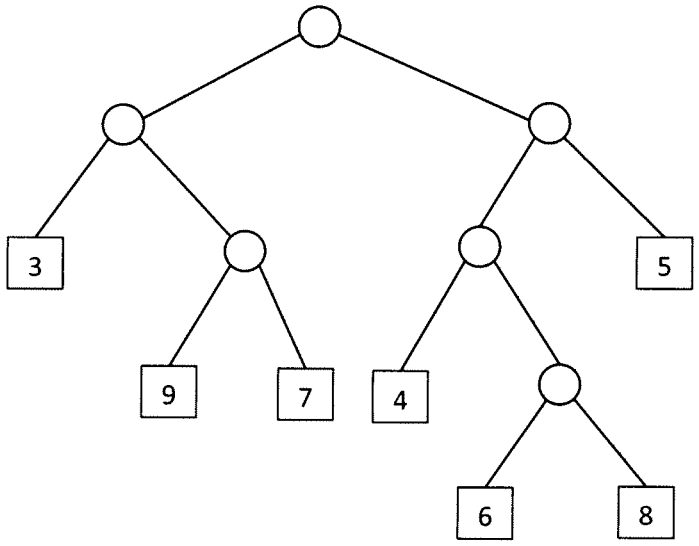
(3d) What will be the base case for your recursive algorithm, i.e., how will you know when to stop recursing? In slot (i) above, write an expression in terms of x to test if x is a base case.

(3e) What should your function do for the base case? Write your answer in slot (ii) above.

(3f) The recursive (non-base) case is handled by the else clause. What should your function do in this situation to make a little bit of progress? Write your answer in slot (iii) above.

(3g) Now it's time for the recursion. Write the recursive call in slot (iv) above.

4. (25 points) For this problem we will work with binary trees that have integer values as the terminal (leaf) nodes, and no missing children (every non-terminal node has exactly two children). We can represent such trees using nested lists. For example, the tree



can be represented by the `list [[3, [9, 7]], [[4, [6, 8]], 5]]`. A tree consisting of a single node would be represented by the integer value it holds.

(4a) Let us write a recursive function to calculate the depth of a tree. The depth is the longest path from the root to any leaf node. The depth of the example tree above is 4, since that is the depth of the leaf nodes 6 and 8. Recursively, the depth of a tree that is not a leaf is 1 plus the depth of either the left child or the right child, whichever is greater. Fill in the missing parts of the function.

```
def tree_depth(x)
    if x.kind_of?(Integer)  then          # base case
      return 0
    else
      left_depth = tree_depth(x[0])
      right_depth = tree_depth(x[1])
      return 1 + max(left_depth, right_depth)
    end
 end
```

(4b) Let us write a recursive function to count the non-terminal nodes (nodes that have at least one child) of a tree. The example tree above has 6 non-terminal nodes. Fill in the missing parts:

```
def count_nonterminal(x)
    if x.kind_of?(Integer) then          # base case
      return 0
    else
      left_count = count_nonterminal(x[0])
      right_count = count_nonterminal(x[1])
      return 1+left_count + right_count
    end
 end
```