

15110 PRINCIPLES OF COMPUTING – EXAM 2B – FALL 2012

Name Answer Key Section H-N

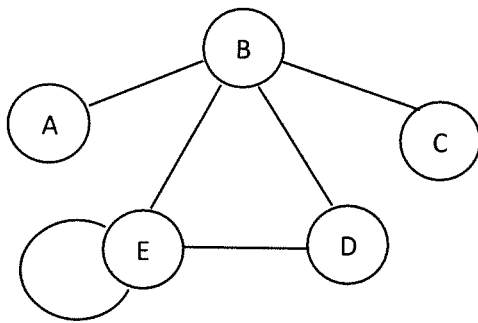
Directions: Answer each question neatly in the space provided.

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

1	<u>25</u>
2	<u>30</u>
3	<u>20</u>
4	<u>25</u>
TOTAL	<u>100</u>

1. (25 points) Recall that graphs can be represented using an adjacency matrix, which in Ruby is an array of arrays.

(1a) Write the adjacency matrix for the following undirected graph to the right of the diagram:



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

(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

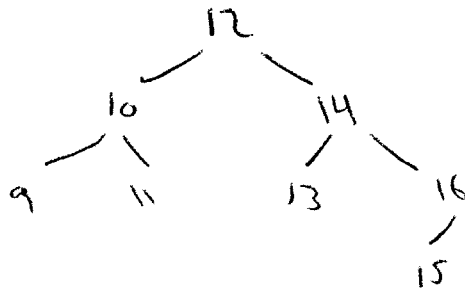
(1c) What is the maximum possible number of links in an directed graph of N nodes? N^2

(1d) The minimum degree of a graph is the smallest number of links of any node. Given an adjacency matrix representation of an unweighted graph, describe an algorithm to calculate the minimum 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. Return the minimum count.

(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 9, 10, 11, 12, 13, 14, 15, and 16. Note that a binary tree must satisfy a specific property for it to be a binary search tree.

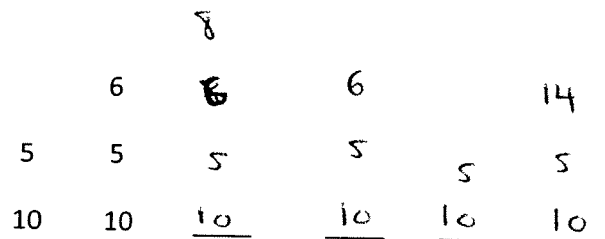


2. Data representation (30 points)

(2a) Suppose a hash table with N buckets has roughly the same number of elements in each bucket and that computing a hash function takes constant time. What is the average runtime cost for accessing an element in this hash table? Write your answer using the big O notation.

$O(1)$

(2b) Consider the following stack that stores numbers. Initially, the stack has 10 at the bottom and 5 at the top. Show the result of executing the operations push(6); push(8); pop; pop; push(14) in the given order. Show the state of the stack at each step. We have given the first step.



(2c) Consider converting the 5 bit binary number 11001 (base two) to decimal form (base ten). Fill in the blanks in the given equation where the expression on the right hand side represents the decimal value.

$$11001_{(2)} = (1 \times 2^{\frac{4}{16}}) + (1 \times 2^{\frac{3}{8}}) + (0 \times 2^{\frac{2}{4}}) + (0 \times 2^{\frac{1}{2}}) + (1 \times 2^{\frac{0}{1}}) = \underline{25}$$

(2d) How many distinct values can be represented using a 4 bit binary number? 16

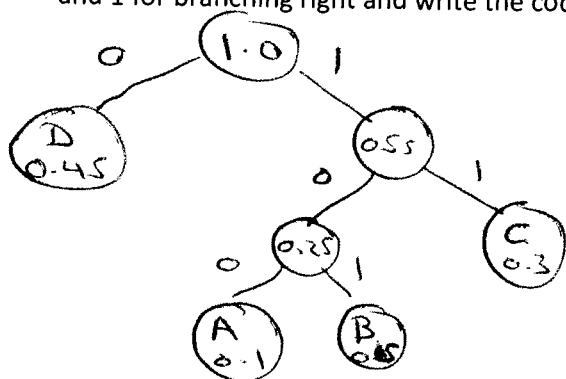
(2e) Suppose that 4 bits are used to represent signed integers using 2's complement representation. Write the binary representation of the number -6 according to this representation. Hint: Note that adding a positive number x to a negative number $-x$ should result in a binary number with 0000 in the rightmost 4 bits.

$$\begin{aligned} 6 &= 0110 \\ 1's \text{ comp} &= 1001 \\ 2's \text{ comp} &= 1010 = -6 \end{aligned}$$

(2f) How many bits are needed to store an uncompressed Red-Green-Blue (RGB) image from a 2 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.

$$2^{20} \text{ pixels} \times 3 \text{ bytes/pixel} \times 8 \text{ bits/byte} = 2^{21} \times 3 \times 2^3 = 3 \times 2^{24} \text{ bits}$$

(2g) Suppose that we are compressing text composed from the letters A, B, C, and D with frequencies 10%, 15%, 30%, and 45%, respectively. Construct a Huffman tree for this text using 0 for branching left, and 1 for branching right and write the code for each letter.



A = 100
B = 101
C = 11
D = 0

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

(2h) Suppose that we have a 10 letter text string consisting of letters A, B, C, D with frequencies given in part (g), which we encode using the normal fixed length ASCII code of 7 bits per letter. What is the compression ratio obtained by Huffman encoding? Answer: 18 divided by 70.

Letter	Freq	Bits	Expected Length
A	0.1	3	0.3
B	0.15	3	0.45
C	0.3	2	0.6
D	0.45	1	0.45
			<u>1.8 bits</u>

Huffman: 10 letters \times 1.8 bits = 18 bits
ASCII: 10 letters \times 7 bits = 70 bits

3. (20 points) Suppose we want to write a recursive function `print_left_nested(x)` to print every non-nil element of a left-nested list, i.e., a list such as `[[[[[nil, 9], 5], 1], 4], 1], 3]`, in the reverse order. For the list in the example, you need to print the numbers 3, 1, 4, 1, 5, and 9. First read (3a) – (3g) before writing any answers.

```
def print_left_nested(x)
  if (i) x.nil? then
    (ii) return nil
  else
    (iii) puts x[1]
    (iv) print_left_nested(x[0])
  end
end
```

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

(3a) What is `x.length`? 2

(3b) What is `x[0]`? [[[[[nil, 9], 5], 1], 4], 1]

(3c) What is `x[1]`? 3

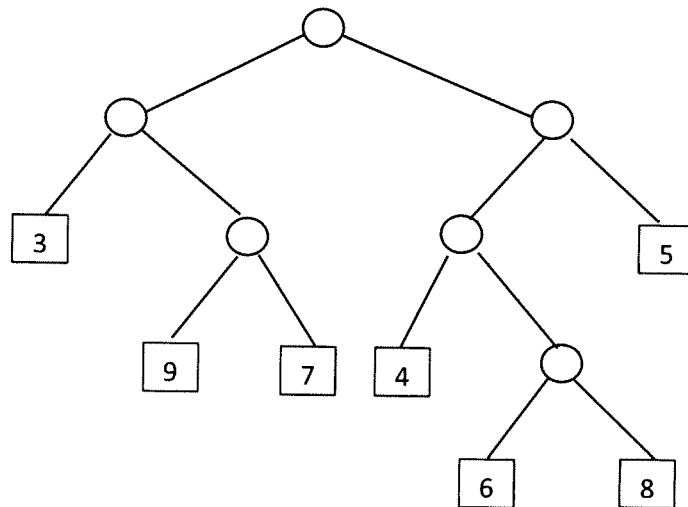
(3d) What will be the base case for your recursive algorithm, i.e., how will you know when to stop recursing? In slot (i) above, write an expression in terms of `x` to test if `x` is a base case.

(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 number of leaves of a tree. The example tree above has 7 leaves. Fill in the missing parts:

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