15-103 PRACTICE PROBLEMS SINCE MIDTERM

Use these problems to get an idea about the format for the actual exam and the level of difficulty of the exam questions. The actual exam may or may not have problems that look like these problems, but the amount and difficulty of the problems will be similar. Note that there are additional topics discussed in class that you are responsible for that do not show up in these practice problems.

1. Consider the following lists define in a functional language like Scheme:

```
(define alice (list 'Pittsburgh 22 'Walk))
(define bob (list 'Monroeville 35 'Bus))
(define carol (list 'Robinson 40 'Car))
(define dan (list 'Cranberry 38 'Car))
(define ethel (list 'Etna 61 'Bus))
(define fred (list 'SquirrelHill 56 'Bike))
```

Each list contains a person's hometown, age and means of commuting to work.

(a) Using Scheme, how would you display Ethel's means of commuting?

(b) Consider the following Scheme function `mysteryfunction` that requires one parameter, a list named `people`:

```
(define (mysteryfunction people)
  (if (null? people)
      0
      (+ 1 (mysteryfunction (rest people)))
  )
)
```

What does the function `(null? people)` mean?

(c) What is the output for the Scheme instruction below using the list definitions above and the function `mysteryfunction` from part (b)?

```
(mysteryfunction (list alice bob carol dan ethel fred))
```

(d) Is the mystery function in part (b) recursive? Why or why not?

2. Consider the following recursive function to compute the number of ways to form combinations of N things taken R at a time:

```
C(N,R) = 1   if R = N
C(N,R) = N   if R = 1
C(N,R) = C(N-1,R) + C(N-1,R-1)   if 1 < R < N
```

(a) Draw the recursive tree that shows all the computational steps required to run this algorithm for N = 4 and R = 2.

(b) Using your tree, what is the value of C(4,2)?

(c) Would dynamic programming help reduce the number of recursive steps in this computation in general? Explain.

3. A Huffman tree is derived for an alphabet of {A,B,C,D,E} as follows:

```
     w
    / \
   x   E
  / \
 A   y
    / \
   B   z
      / \
     C   D
```

Each left branch is encoded with a 0 and each right branch is encoded with a 1.

(a) Decode the following message using the Huffman tree above.

`101010111100011010011010`

(b) If we encoded this alphabet using a minimal fixed-length encoding scheme, how many bits would we use to encode each letter? Why?

(c) Complete the EBNF rules below for any valid message derived from the Huffman tree above.

message → ______________________________ (one or more letters)

letter → _________________________________ (binary pattern for A or binary pattern for B or ... *etc.*)

4. The following matrix represents a weighted directed graph with 6 nodes labeled U, V, W, X, Y, Z.

	U	V	W	X	Y	Z
U	0	7	4	0	0	0
V	0	0	0	6	5	0
W	0	0	0	8	3	0
X	0	0	0	0	0	2
Y	0	0	0	0	0	9
Z	0	0	0	0	0	0

(a) Draw a picture of the graph represented by the matrix above.

(b) Let L(i) represent the cost of the shortest path from node i to node Z in the graph. Find L(i) for each node of the graph using the Shortest Path algorithm discussed in class.

(c) Consider a greedy algorithm that builds a path to node Z by starting at node U and taking the lowest cost edge to another node. From that node, we take the lowest cost edge again. We repeat this until we reach node Z. Explain using this graph why this greedy approach does not yield the shortest path from U to Z.

5. The following algorithm is designed to compute x * y through repeated addition, assuming x > 0 and y > 0.

```
1.  Set result = x
2.  Set i = 1
3.  While i ≠ y do the following:
    (a) Add 1 to i
    (b) Add x to result
4.  Output result
```

The invariant for this loop is "result = x * i". Show that this algorithm is correct by giving logical explanations for the following steps.

(a) Show that the invariant must be true immediately before the loop begins.

(b) Argue that if the invariant is true at the beginning of each iteration of the loop, it must be true at the end of each iteration as well.

(c) Argue that if the loop exits, the output value must be x * y.

(d) Explain why the loop must exit.

6. Consider the following algorithm that computes the sum of the first n non-negative powers of 2: $2^0 + 2^1 + ... + 2^{n-1}$, where $n \geq 1$.

```
1. Input n (assume n ≥ 1)
2. Set sum to 1
3. Set value to 2
4. Do the following n-1 times:
   a. Add value to sum
   b. Multiply value by 2
5. Output sum
```

(a) Counting assignments ("Set ..."), additions and multiplications as operations, exactly how many operations does the algorithm perform in terms of the variable n?

(b) What is the order of complexity of this algorithm in big O notation? Explain your answer.

(c) Use induction to show that the sum that is computed in the algorithm is also equal to $2^n - 1$, $n \geq 1$. (HINT: Show that the sum is $2^n - 1$ when n = 1. Then show that if the sum is $2^n - 1$ for some n, then the sum = $2^{n+1} - 1$ for n+1.)

7.

(a) Determine a set of boolean values that can assigned to the four variables in the following logical sentence so that it is satisfied (i.e. it evaluates to *true*), or explain why no assignment is possible.

(A v B) & (C & ~D) & (A --> C)

(b) If a logical sentence has N boolean variables, at most how many boolean assignments need to be tested for a solution to this satisfiability problem? Explain.

(c) What does your solution to (b) say about the computability of these types of problems?

8. Let E be a program that takes some input program P and determines if P will halt for at least **1** possible input value when it is run. (For example, imagine that E takes one of your 15-111 programs and determines if the program can terminate successfully for at least one input data value.)

(a) Explain why this problem is classified as a "decision problem".

(b) Let F be a program that works as follows with an input program P:

```
Run E with program P as its input.
If E responds that P will halt for at least
one data input:
     go into an infinite loop
Otherwise:
     output DONE
```

What happens if F is executed with itself as input?

(c) What does the result of (b) say about the existence of program E?