Tim Cheeseman
CS 260-003
Summer 2014
Assignment 3 Sample Solution

3.1)

a) D, F, J, K, L, M, N
b) A
c) A
d) F, G, H
e) A, B, E
f) E, I, M, N
g) D’s right sibling is E, E has no right sibling
h) Left: B, D, E, F, I, M, N
Right: H, L
i) 1
j) 2

3.2)

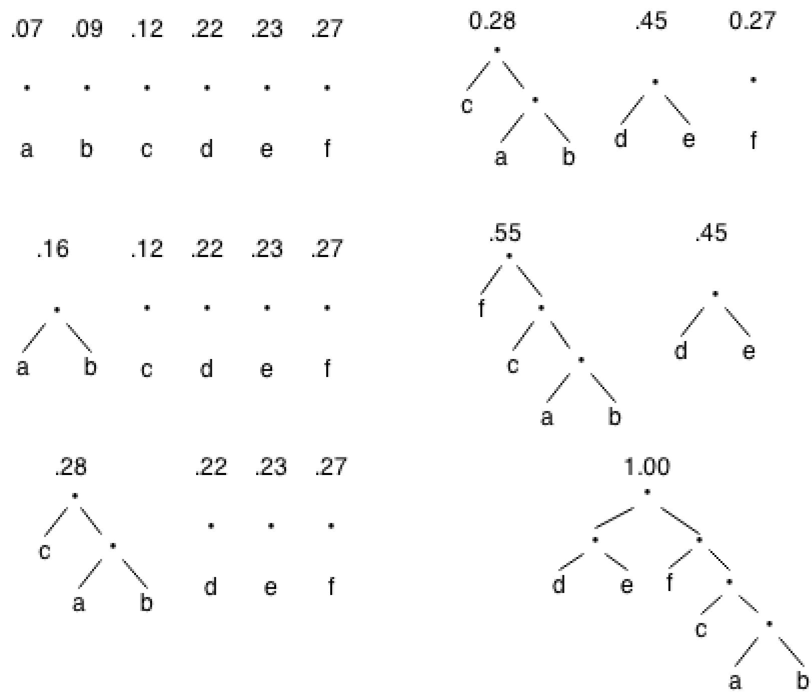
There are 6 different paths of length three:

- A-B-E-I
- B-E-I-M
- B-E-I-N
- A-C-G-J
- A-C-G-K
- A-C-H-L

3.6)

	preorder(n) < preorder(m)	inorder(n) < inorder(m)	postorder(n) < postorder(m)
n is to the left of m	✓	✓	✓
n is to the right of m			
n is a proper ancestor of m	✓	✓	
n is a proper descendant of m		✓	✓

3.20)



Character	Code	Code Length	Probability
a	1110	4	.07
b	1111	4	.09
c	110	3	.12
d	00	2	.22
e	01	2	.23
f	10	2	.27

Average code length = $4 \cdot 0.07 + 4 \cdot 0.09 + 3 \cdot 0.12 + 2 \cdot 0.22 + 2 \cdot 0.23 + 2 \cdot 0.27 = \mathbf{2.44}$

3.21)

Assume depth(a) > depth(b) and P(b) < P(a)

Since P(b) < P(a), then b will always be chosen to be combined before a. If it is chosen to be combined with a, then depth(a) > depth(b) will not be true. If it is first combined with any symbol, then that resulting subtree will always have a depth greater than any nodes added onto it in the future (including a), so depth(a) > depth(b) will not be true. Therefore, when depth(a) > depth(b), P(b) >= P(a).

Implementation)

fib_rec(n):

$T(n) = T(n-1) + T(n-2) + O(1)$
$T(1) = T(2) = O(1)$

$T(n) = O(2^{n-1}) + O(2^{n-2}) + O(1) = O(2^n)$

Note: $O(2^n)$ is sufficient for this answer, though this is not a tight bound. You'll notice that T(n) and fib(n) have the same recursive definition. If you investigate, you'll find that the tight bound is $\Theta(\varphi^n)$ where φ is the golden ratio (≈1.618).

fib_memo(n):

The worst-case running time of fib_memo(n) is the same as fib_rec(n). The benefit, however, is after running fib_memo(x), then the running time of fib_memo(y) where y <= x is O(1).