Joseph Mulray

HW3: Implimentation

January 30, 2017

Problem 1:

Include with this portion of your homework a copy of your two Fibonacci programs, including your memoisation function, from the implementation section.

Fib 1:

```
def fib(num):
    #fib function passes in num from arugment

    #let Fib(0)=1, and Fib(1)=1.
    if num <= 1:
        return 1

    #use recursion to calculate the fib numbers
    return fib(num - 1) + fib(num - 2)
```

Fib 2:

```
def fib2(num):

    #let Fib(0)=1, and Fib(1)=1.
    if num <= 1:
        return 1

    # if memolist[num] exists: return that number
    if memolist[num]:
        return memolist[num]

    #does not exist populate that number
    memolist[num] = fib2(num - 1) + fib2( num - 2)

    #return value of that number
    return memolist[num]
```

The first fib without memoisation, the time complexity is *O(2^n)* for each iteration of fib called will result in the following:

fib(num - 1) + fib(num - 2)
(num - 2) + (num -3)
...
will keep increasing until *n <= 1*

*2*2*2*2 = 2^n*

Incrementing by 2* each time, resulting in a time complexion of *O(2^n)*

For fib with memoisation, the time complexity will be *O(n)* linear because once a Fibonacci number is calculated it is stored into an array where the index is searchable, making the worst case scenario is n times.

Problem 2:

If the input size was also unbounded, the time complexity would still be the same, linear because you will always have a list size to start from, reguardless of the size of n, making it still *O(n)*

Problem 3.1

a. D, M, N, J, K, L
b. A
c. A
d. F, G, H
e. B, A
f. I, M, N
g. D: E, E: has no right sibling
h. F is to the left, H is to the
i. C has a depth of 2
j. C has a height of 2

Problem 3.2

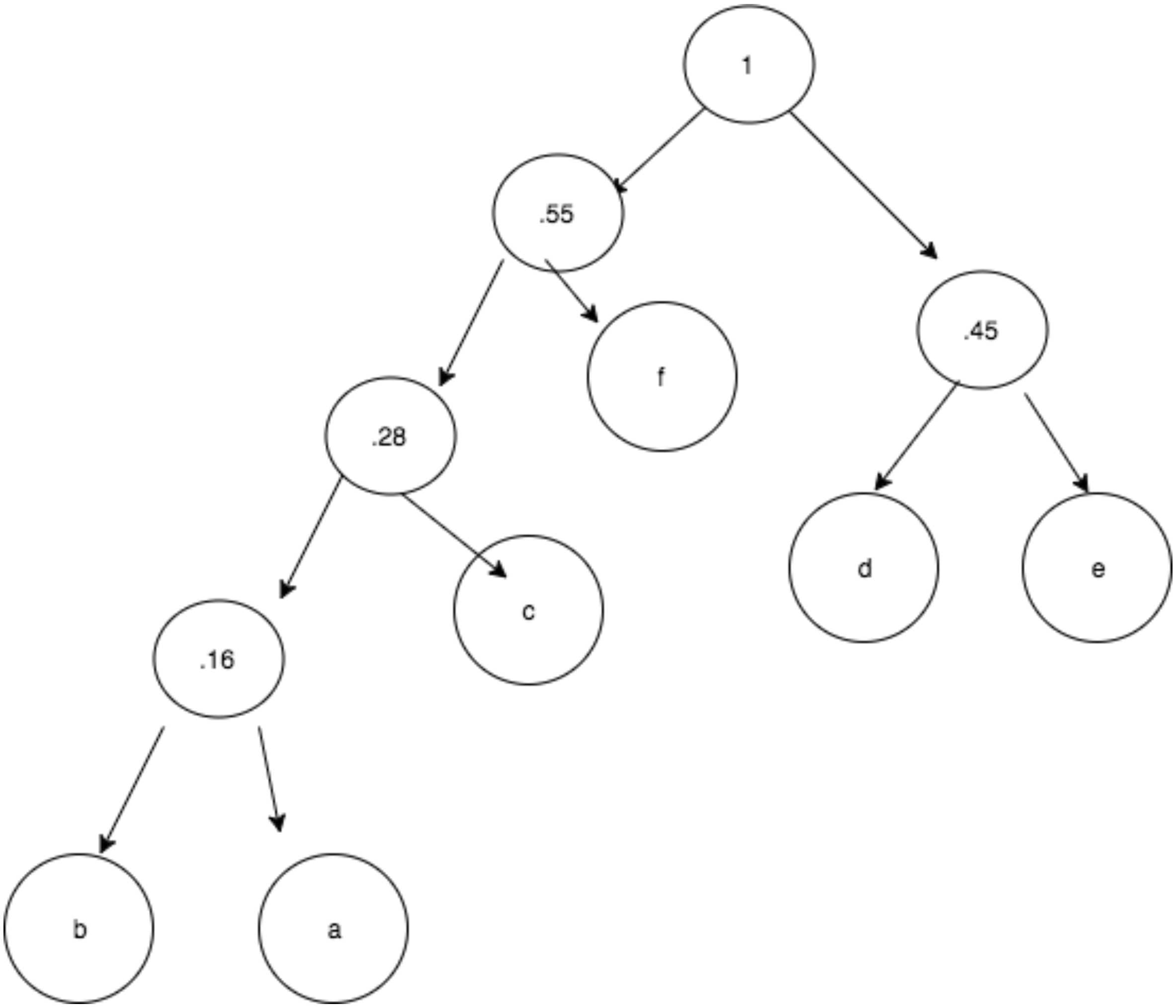
There are 6 paths of length 3:

- ABEI
- BEIM
- BEIN
- ACGJ
- ACHL
- ACGK

Problem 3.6

	pre(n) < pre(m)	**in(n) < in(m)**	**post(n) < post(m)**
left of m	Y	Y	Y
right of m			
ancestor of m	Y	Y	
descendant of m		Y	Y

Problem 3.2:



Character	Probability	Code	Weight
a	.07	11111	5
b	.09	11110	5
c	.12	1110	4
d	.22	110	3
e	.23	10	2
f	.27	0	1

(1).27 + (2).23 + (3).22 + (4).12 + (5).09 + (6).07

Average length = 2.74

Problem 3.21:

Prove that the probability of symbol b is no less than that of a:

Since that A > B depth wise, then the probability is B > A because the more depth you have the lower the probability, and since A has a greater depth that you can assume that B has a greater than or equal probabilty than A.