Romeo Capozziello HW 2

Problem 1:

A binary search of a sorted array with 3 unique elements where the search is always successful can be viewed as an array with each element having a probability of being selected = 1/3 and the number of comparisonsa binary search requires to find the element is shown inside the array.

2	1	2

Then the averagenumber of comparisons would be A(3) = (1/3)[2 + 1 + 2] = 5/3
Find the A(7) for an array with 7 elements for the same binary search.

(a) Draw the 7-element array showing the number of comparisons needed to find each element.

4	3	2	1	2	3	4

(b) Determine A(7)

A(7) = (1/7)[4 + 3 + 2 + 1 + 2 + 3 + 4] = (1/7)(19) = 19/7

(c) Using what you learned from parts (a) and (b), find the summation expression for A(n). To simplify your work, assume that n = 2^k–1. Use k in your summation. Leave your answer in closed form in terms of n.

n = 2^k - 1, A(n) = $\frac{1}{n}(1+\sum_{i=2}^{k+1}$ 2i)
Ex: n = 7, 7 = 2^3 - 1, k = 3, A(7) = $\frac{1}{7}(1+\sum_{i=2}^{3+1}$ 2i) = 19/7

(d) Simplify your a(n) for large values of n.

$\sum_{i=1}^{k} i = \frac{k(k+1)}{2} \rightarrow \sum_{i=2}^{k} i = \frac{k(k+1)}{2} - 1 \rightarrow \sum_{i=2}^{k+1}$ i = $\frac{(k+1)(k+2)}{2} - 1$

$\rightarrow \sum_{i=2}^{k+1}$ 2i = $\frac{2(k+1)(k+2)}{2} - 1 = (k+1)(k+2) - 1$

A(n) = $\frac{1}{n}(n(n+1) - 1) = (n+1) - \frac{1}{n}$

Problem 2:

Consider the algorithm below
Precondition: n is a non-negative integer

```
function f(n)
{
    temp = 0
    if (n != 0)
    {
        for (i = 1; i ¡= 3; i++)
            temp = temp + n * f(n-1)
        return temp
    }
    else
        return 1;
}
```

Solvefor the closed form by repeated substitution

Answer

Problem 3:

The recurrence relation is given as

$a_n = 2a_{n-1} + 3a_{n-2}$

Use the method linear homogeneous characteristic roots to solve for the closed form, with given initial conditions

$a_0 = 2$ and $a_1 = 4$

(a) Find the general solution

Answer

(b) Find the Specific solution

Answer

(c) Use your closed form result to find a_s

Answer

Problem 4:

Two average complexities are given below.
Variable p is probability value in the interval of [0.0, 1.0]. Variable n is the problem size.

$A_1 = p(n^2 + 1) + (1 + p)3n$
$A_2 = (1 - p)(3n^2 + n) + p(2n)$

(a) Determine the condition for which A_1 is faster than A_2.

Answer

(b) Approximate the range of p values for which A_1 is faster than A_2 for large values of n.

Answer

(c) Given n=5,find the range of values for p where A_2 is faster than A_1.

Answer

(d) Can you determine the problem sizen where A_1 is always faster? Why or why not?

Answer