Algorithms and Data Structures
Homework5

bessefiany

March 8, 2021

1 Problem 5.1

1.1 a

naive recursive:

```
    Fibonacci(n)
        if (n < 2)
            return n
        else
            return Fibonacci (n - 1) + Fibonacci (n - 2)
```

bottom up:

```
    Fibonacci(n)
        if (n = 0 or n = 1)
            return n;
        A[0] = 0
        A[1] = 0
        for i=2 to n:
            A[i] = A[i-1] + A[i-2]
        return A[n]
```

closed form:

$\phi = \frac{1+\sqrt{5}}{2}$

Fibonacci(n)

return $\frac{\phi^n}{\sqrt{5}}$

matrix representation:

1.2 c

closed form gives a different value as for n = 1:
$\frac{\phi^1}{\sqrt{5}} = \frac{1}{2\sqrt{5}} + \frac{1}{2} \approx 1$ but not 1,
when n is high, the error get larger enough to change the values.

2 Problem 5.2

2.1 a

By doing simple calculations, we can not that during a multiplication, a brute force implementation is equal to $\Theta(n^2)$,this is the needed time so each Bit 1 multiplies to bit 2. A supplementary Θ(n) which gonna get deleted as we are only interested in the higher polynomial. We deduce that time complexity for such an operation is $\Theta(n^2)$

2.2 b

We know that the general formula of a number: $y = a.10^{\frac{n}{2}} + b$
where a=left;b=right;n=digit
$A = |L|R| = L.10^{\frac{n}{2}} + R$
$B = |L|R| = L.10^{\frac{n}{2}} + R$
So $A.B = (L_A.10^{\frac{n}{2}} + R_A)(L_B.10^{\frac{n}{2}} + R_B)$
$= L_A L_B.10^n + L_A R_B 10^{\frac{n}{2}} + R_A L_B 10^{\frac{n}{2}} + R_A R_B$
$= (L_A L_B.10^n + (L_A R_B + R_A L_B)10^{\frac{n}{2}} + R_A R_B$

Implementing the algorithm:

```
float mult(a,b)
    n=max(bits in a,bits in b)
    if(n==1)
        return a×b
    else
        a=left [n/2] bits of A
        a=right [n/2] bits of A
        a=left [n/2] bits of B
        a=right [n/2] bits of B

        ac=mult(a,c)
        bd=mult(b,d)
        adbc=mult(a+b,c+a)
        return (ac × 10^n + ((adbc − ac − bd) × 10^(n/2)) + bd
```

2.3 c

THE multiplication function was done 3 times where $\frac{n}{2}$ is the parameter:
$T(n) = 3T(\frac{n}{2}) + \Theta(n)$

d)

$\rightarrow n$

$\rightarrow \frac{3}{2} n$

$\rightarrow \left(\frac{3}{2}\right)^2 n$

$h = \log_2 n$

$$\sum_{i=0}^{h} \left(\frac{3}{2}\right)^i n = n \sum_{i=0}^{\log n} \left(\frac{3}{2}\right)^i = n \left[\frac{1\left(1 - \left(\frac{3}{2}\right)^{\log n}\right)}{1 - \frac{3}{2}} \right]$$

$$= -2n \cdot \left(1 - n^{\log_2 \left(\frac{3}{2}\right)}\right)$$

$$= -2n + 2n \cdot n^{0.58}$$

$$= 2n^{1.58} - 2n$$

So $T(n) = \Theta(n^{1.58})$

e) according to master theorem

$a = 3$

$b = 2$

$f(n) = n^{\log_2 3} = n^{1.58}$

using case I of master theorem

$f(n) = O(n^{\log_2 3 - 0.08})$ for $\epsilon = 0.08$

$= O(n^{\log_2 3}) = \Theta(n^{1.58})$

which is better for implementation of

$= n^2$