Faculty of Computers and Information

Cairo University

Midterm Exam

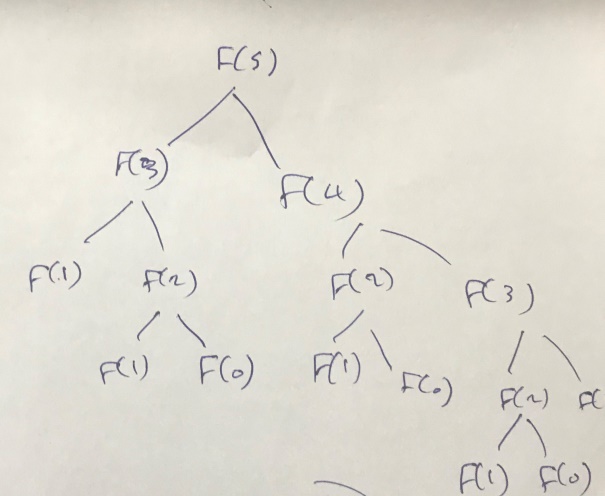
|  |  |  |
| --- | --- | --- |
| Department: CS |  |  |
| Course Name: Algorithms Analysis and Design |  | Date: 17/11/2019 |
| Course Code: | Duration: 1 hours |
| Examiner(s): Dr/ Basheer Youssef | Total Marks: 45 |

Q1 (9 points)

Given the following recursion function:-

F (n) = n when n >=0 and n <2

F (n) = F (n-1) + F (n-2) when n>=2

Solve this problem in three different ways such as what was explained in the lectures and determine what its complexity. And compute f(5) by each methods and show the steps of your answer.

**Answer each answer 3 point in each answer 1n point foe order, code and drawing**

1. Recursion O(2n)

fib(int n)

{

if( n <= 1 && n.=0) return n;

else return fib(n-1) + f(n-2)

}

F(0) = 0; F(1) = 1, F(2) =1 , F(3) = 2 , F(4) = 3 , f(5) = 5

student must how it in recursion tree as the following

1. Dynamic programming O(n) make array and store in it the value from 0 to n and use previous to cell to compute the next one

Fib(int n)

{

int X[n]; X[0]= 0; X[1]=1;

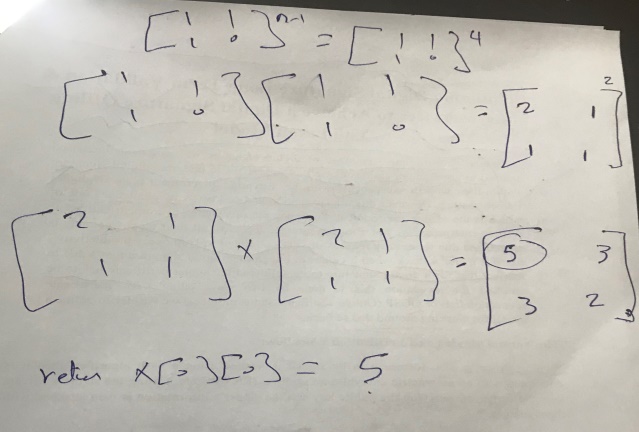
for (int i =2 ;i<=n ;i++)

X[i]= X[i-1] + X[i-2] ;

Return X[n];

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 2 | 3 | 5 |

1. Recursive squaring O(lgn)

Fib(n)

{

Int x[2][3];

x[0][0]=1; ]; x[0][1]=1; x[1][0]=1; x[1][1]=0;

Recsq(x,n-1);

Return x[0][0];

}

Q2 what is the time complexity T (n) of the nested loops below? For simplicity, you may assume that n is a power of 4. That is, n = 4k for some positive integer k. [4 points]

for (i = 1; i <= n; i++) {

j = n;

while (j > = 1){ < body of the while loop > //Needs Θ (1).

j = [j/4]; } }

**Answer (lg4n)**

Q3 Prove that (9 points)

1. worst case of quick sort = n2 using recursion tree



2. What is the complexity if the pivot split list is always 1/10 and 9/10? using recursion tree



3. Suppose we alternate lucky, unlucky, lucky, unlucky, lucky, ….

L(n)= 2U(n/2) + Θ(n)lucky

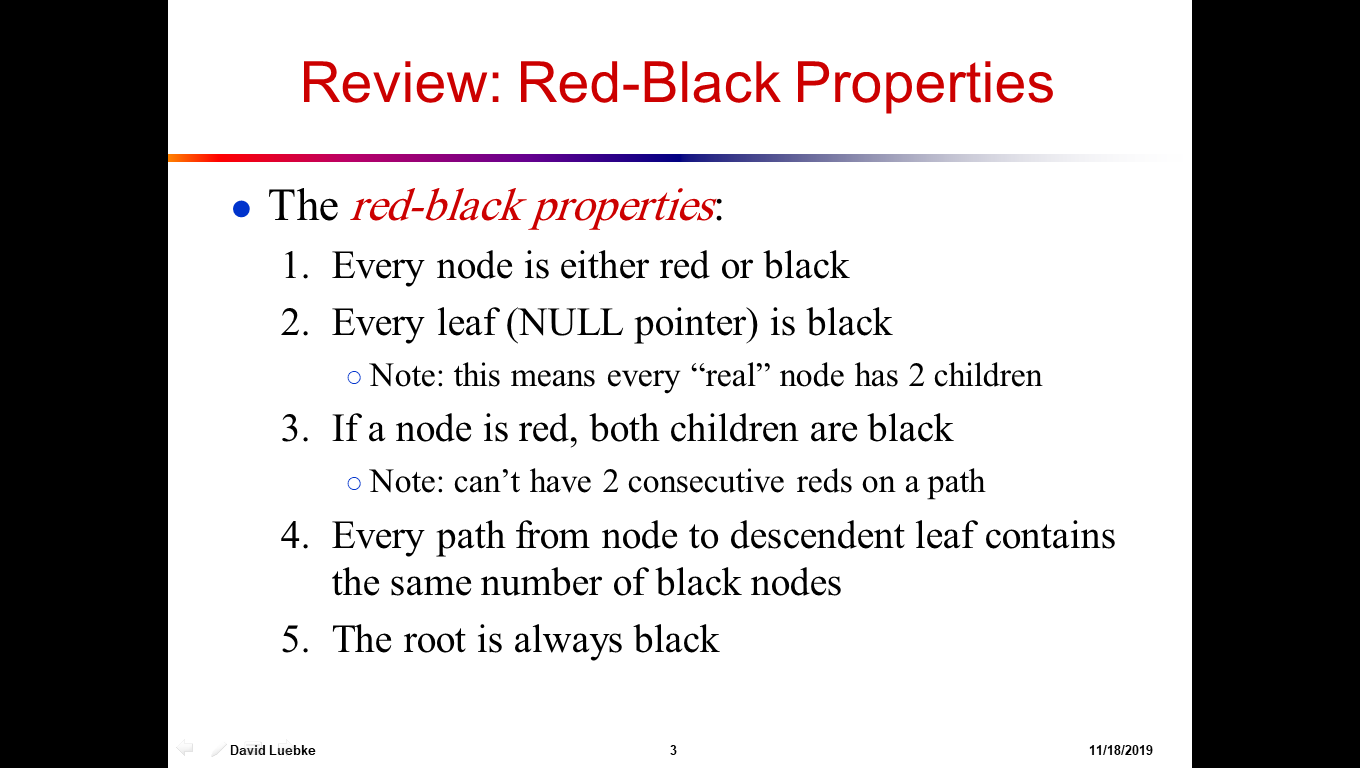
U(n)= L(n –1) + Θ(n)unlucky

What is the time complexity in this case.



Q4 what is properties of Red Black tree and how can you get the maximum of tree and predecessor of any key, write a pseudocode for these functions ? (8 points)

**Answer**



Get maximum

From root go right until reach null child and return thr value of most right node O(lgn)

Get predecessor of any node

From this go to left one and get maximum of this sub tree

Q5 Using master method to solve (9 points)

1. T(n) = T(n/2) + n (3 points) **answer n**
2. T(n) = T(n/2) + n0 (3 points) **lgn**
3. T(n) = 4T(n/4) + n(3 points) **nlgn**

Q6 Using count sort to sort the following data (8 ,5,7,3,6,1,2) (6 points)

A

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **8** | **5** | **7** | **3** | **6** | **1** | **2** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **index** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Occurence** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** |
| **Acumelation** | **1** | **2** | **3** | **3** | **4** | **5** | **6** | **7** |
| **final** | **0** | **1** | **2** | **3** | **3** | **4** | **5** | **6** |

C

B

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **5** | **6** | **1** | **8** |