Daniel Campos

October 4th, 2012

CSCI-2300 Section 5

HW 3

1) Problem 2.13

A)

3 Vertices =1

5 Vertices =2

7 Vertices =5

For even values there cannot be a  since it cannot exist. Since to have a full tree it is necessary to have a full tree(2 children or none) all even values would produce trees with at least one leaf with only one child, therefore not being full trees.

B)  or



2)Problem 2.14

function remove-duplicate(a[1...n]

Input: An array of numbers a[1...n]

Output: Array A with duplicates removed

Construct an array temp[1..n]:

temp[i] has two fields key and value

for i = 1 to n

temp[i].value = a[i]

temp[i].key = i

sort temp based on value

remove duplicates from temp based on value:

keep the entry with minimum key

sort temp based on key

construct array A from temp:

A[i] = temp[i].value

return A

3) Problem 2.17

In C++

int find\_integer(int A)// modified form of binary search

{

int upper=A.size()-1;//Initializing space counters

int lower=0;

int mid=0;

while((upper-lower)>1)//will loop along as distance between two points is

{ //greater than 1

mid=(upper+lower)/2 //middle is total over 2

if(A[mid]==mid)//Condition Found

{ return mid;}

else if(A[mid]<mid)// if the value is greater than mid then mid

{ lower=mid;}// becomes lower bound

else //value is lower than mid so mid becomes upper

{upper=mid;}

}

return 0;// item was not found

}

4) Problem 3.1

|  |  |  |
| --- | --- | --- |
| Edge | Pre Number | Post Number |
| A | 1 | 12 |
| B | 2 | 11 |
| C | 3 | 10 |
| D | 13 | 18 |
| E | 5 | 6 |
| F | 4 | 9 |
| G | 14 | 17 |
| H | 15 | 16 |
| I | 7 | 8 |

Tree Edges: (A,B), (B,C,), (C,F),(F,E),(F,I),(D,G),(G,H)

Back Edges: (E,A,),(H,D)

5) Problem 2.23 B

Function get\_major\_element(a[1,….,n])

Input: an array, A, of objects

Output: majority of elements of A

If n==2

If a[1]==a[2] return a[1]

Else return No Major Element

Create a temporary array b

For i=1 to n

If a[i]== a[i+1]

Insert a[i] into b

i=i+1

return get\_major\_element(b)

The worst case run time is t(n)=T(n/2)+O(n) =O(n)

Let m> n/2 be the number of copies of the majority element

Let p be the number of matching pairs of the majority element

The number of matching copies of the majority element that pass to the next recursive level is p. n-m elements are not the majority element. M-2p copies of the majority element are not in pairs and therefore eliminates one other element each. Therefore, the maximum numbers of matching pairs that aren’t the majority element are  If n is even and if n is odd, both of which are less than p.

6) Problem 3.3 (ignore)

A)

|  |  |  |
| --- | --- | --- |
| Node | Pre Number | Post Number |
| A | 1 | 16 |
| B | 3 | 4 |
| C | 2 | 15 |
| D | 5 | 14 |
| E | 6 | 8 |
| F | 7 | 13 |
| G | 9 | 10 |
| H | 11 | 12 |

B) Sources: A and B

Sinks: G and H

C)Topological Ordering:

A,D,G,C,F,B,E,H(Visual left to right top to bottom)

A,B,G,H, D,E,C,F(Fewest Edges First)

(largest number of available vertex first

D)How many topological ordering does the graph have: