CPS 130 Exam 2

Spring 2000

12:40-1.55pm, Tuesday April 18th
Closed book exam

NAME:_____

Problem	Max	Obtained
1	20	
2 (a)	5	
2 (b)	25	
3 (a)	25	
3 (b)	25	
Total	100	

Comments:

- You can use any of the algorithms covered in class without describing it.
- When asked to describe an algorithm it is completely ok to do so with words (and few accompanying pictures if it helps the description).

HONOR CODE

I have obeyed the honor code.

SIGNATURE:_____

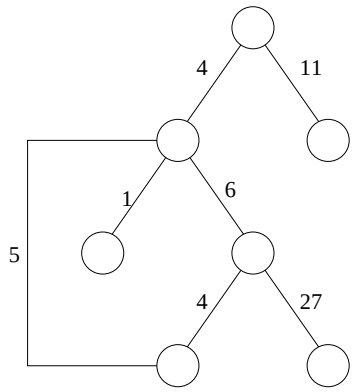
[*20 points*] **Problem 1:**

As we have seen, a red-black tree T supports the operations Insert(T, x), Delete(T, x), and Findmin(T) in $O(\log n)$ time. Thus, we can use T to sort an array A of n elements in $O(n \log n)$ time with the following algorithm:

```
for i=1 to n do
  Insert(T,A[i])
od
for i=1 to n do
  a[i]=Findmin(T)
  Delete(T,a[i])
od
```

Describe a modification of the three red-black tree operations such that if A contains only k different elements, the above algorithm sorts the n elements in $O(n \log k)$ time.

[*30 points*] **Problem 2:** Consider an undirected weighted graph which is formed by taking a binary tree and adding an edge from *exactly one* of the leaves to another node in the tree. We call such a graph a *loop-tree*. An example of a loop-tree could be the following:



Let n be the number of vertices in a loop-tree and assume that the graph is given in the normal edge-list representation without any extra information. In particular, the representation does not contain information about which vertex is the root.

1. How long time would it take Prim's or Kruskal's algorithms to find the minimal spanning tree of a loop-tree?

2. Describe and analyze a more efficient algorithm for finding the minimal spanning tree of a loop-tree. Remember to prove that the algorithm is correct.

[*50 points*] **Problem 3:**

Consider the numbers $(A_n)_{n>0} = (1, 1, 3, 4, 8, 11, 21, 29, 55, \ldots)$ defined as follows:

$$\begin{aligned} A_1 &= A_2 = 1 \\ A_n &= B_{n-1} + A_{n-2} \qquad n > 2 \\ B_1 &= B_2 = 2 \\ B_n &= A_{n-1} + B_{n-2} \qquad n > 2 \end{aligned}$$

A_n can be computed using the following recursive procedures:

```
ComputeA(n)
   if n<3 then
      return 1
   else
      return ComputeB(n-1)+ComputeA(n-2)
   fi
end

ComputeB(n)
   if n<3 then
      return 2
   else
      return ComputeA(n-1)+ComputeB(n-2)
   fi
end
```

1. Show that the running time $T_A(n)$ of ComputeA(n) is exponential in n.
   (*Hint:* Show for example that $T_A(n) = \Omega(2^{n/2})$)

2. Describe and analyze a more efficient algorithm for computing A_n.