COS 423 Problem Set 3 Due Wed. 2/25/2009
Spring 2009

No collaboration

1. Prove that for any rank-balanced tree there is an assignment of colors that makes it a red-black tree. (Recall from CLRS Chapter 13 that in a red-black tree every node is either red or black, the root and the missing nodes are black, every red node has black children, and every path from the root down to a missing node has the same number of black nodes. Thus from any node any path down to a missing node has the same number of black nodes.) Give an example of a red-black tree such that there is no way to assign ranks to make it into an rb-tree.

2. In the following, $p(x)$ denotes the parent of node x; "rotate edge $(x, p(x))$" means do a left (right) rotation at $p(x)$ if x is the right (left) child of $p(x)$. This makes $p(x)$ the left (right) child of x. Consider the following variant of splaying:

```
vsplay(x): while x is not the root do begin
                if x and p(x) are both left children or both right children then
                     rotate edge (p(x), p(p(x)));
                rotate edge (x, p(x));
           end
```

Each execution of the loop body does one or two rotations. Verify that the optional rotation of edge $(p(x), p(p(x)))$ does not change $p(x)$ and that *vsplay*(x) makes x the root. Prove that the amortized number of rotations needed to do a vsplay at a node x is $O(\phi'(x) - \phi(x)) + 1$ where the potential is the same as that defined in class and $\phi(x)$ is the potential of x before the vsplay, $\phi'(x)$ the potential after. How small can you make the constant factor? Under what circumstances can you avoid the "+ 1"? Note: the paper on self-adjusting search trees posted on the class website calls this version of splaying *simple splaying* and claims a constant factor of approximately 3.16, as compared to 3 for standard splaying. You do not need to obtain 3.16; any constant factor will suffice to solve the problem. But smaller is better.