CPS 230 Homework-2

Write the solution to each problem on a single page of a separate sheet of paper.
The deadline for handing in solutions is October 2nd.

1. Searching Problem: (20 points)


Note that $A$ must have at least one local minimum.

We can obviously find a local minimum in $A$ in $O(n)$ time. Describe a more efficient algorithm for finding a local minimum.

Explain your algorithm, analyze it, and describe why it is correct.

2. Searching Problem 2: (20 points)

Suppose we have a sorted linear array $A[1..n]$ and we search for an item $a$. Give an algorithm that finds $a$ in time $O(\log k)$, where $k$ is the position of $a$ or of the smallest item larger than $a$ in $A$.

[Hint: You have to be faster for items near the beginning of the array so you will have to bias your search towards that region.]

3. Binary Tree Problem: (20 = 8 + 12 points)

(a) Explain the preorder and postorder sequences of a binary tree and give recursive functions that store the position of a node $\mu$ in these sequences in $\mu \to pre$ and $\mu \to post$.

(b) Prove that $\mu$ is a proper ancestor of $\nu$ if and only if $\mu \to pre < \nu \to pre$ and $\mu \to post > \nu \to post$.

4. Red-Black Tree Problem: (20 points)

Consider a red-black tree formed by inserting $n$ nodes. Argue that if $n > 1$, the tree has at least one red edge.

(Notice that we are talking about red-black tree formed by insertion. Otherwise a red-black tree can have all black edges.)

5. Amortized Algorithm Problem: (20 points)

Show how to implement a queue with two ordinary stacks so that the amortized cost of each ENQUEUE and each DEQUEUE operation is $O(1)$.

6. Bonus Problem: (20 points are not included in the hundred percent of credit, but can make up the points you lose in any of the seven homework.)

Show how to implement a dynamic set that efficiently supports the FIFO queue operations ENQUEUE and DEQUEUE, as well as MINIMUM. Implement all three operations in $O(1)$ amortized time.