CSE 5311 Section 003 Fall 2004

Quiz 3

October 6, 2004

This is a closed-book, closed-notes, closed-computer, closed-neighbor quiz.

1. (5 points) Show the final B-tree that results from inserting the keys A, B, C, D, E, F, G, H and I (in this order) into an empty B-tree, where minimum degree $t = 3$.

2. (3 points) Show the B-tree that results from deleting X from the following B-tree, where minimum degree $t = 3$.
3. (2 points) Circle the answer below that is closest to the maximum number of keys that can be stored in a B-tree of height 2 and minimum degree \( t = 512 \).

(a) 1,000,000
(b) 1,000,000,000
(c) 1,000,000,000,000

4. (2 points) Define what it means for a problem to exhibit optimal substructure.

5. (1 point) Besides optimal substructure, what is the other main element of dynamic programming?

6. (2 points) In addition to a problem exhibiting optimal substructure, what two characteristics must a greedy choice possess to yield an optimal solution (i.e., the two aspects of proving the greedy choice property)?
7. Consider the problem of computing Fibonacci numbers, where the Fibonacci number $F(n)$ of $n$ is defined as follows.

$$F(n) = \begin{cases} 
1 & n < 3 \\
F(n - 1) + F(n - 2) & n > 3
\end{cases}$$

Recall from class that the straightforward implementation of a recursive algorithm to compute $F(n)$ is $\Omega(2^n)$. We want to design a $O(n)$ dynamic programming solution to this problem.

(a) (3 points) Draw the binary tree representing the recursive calls made by the recursive algorithm and indicate identical subproblems. The root of the tree is $F(n)$, the root’s two children are $F(n - 1)$ and $F(n - 2)$, and so on. Draw the tree deep enough to identify at least two different, overlapping subproblems.

(b) (2 points) In terms of $n$, how many unique subproblems are there for computing $F(n)$?

(c) (5 points) Give pseudocode for a $O(n)$ algorithm $\text{FIBONACCI}(n)$ that uses a bottom-up, iterative design to compute and return $F(n)$.