Quiz 4 – 20 points possible
5 questions on 3 pages

1. (5 pts) Give the Big-Oh running time for the following operations;

a) Insert into a 2-3 tree

b) Finding the inorder successor of a value in a 2-3 tree

c) Insert into a hash table using separate chaining

d) Search for an item in a hash table using separate chaining (give the best case run time since we use hash tables for their best case and not their worst case)

e) Rehashing all items in a hash table of size N to a hash table of size N^2

2. (2 pts) Which two collision resolution strategies may never find a location to insert an item into a hash table even if there are empty spaces?

3. (3 pts) Assume that you have a hash table of size N and you are using double hashing as your collision resolution strategy. What does a load factor of 1 ($\alpha = 1$) tell you about the table and the items in it? If this situation poses a problem, explain why it is a problem and what you can do to fix the problem?

Continued on next page

4. (5 pts) Show the resulting 2-3 trees if we insert the elements 4, 6, 2, 9, 0, 5, 7, and 3 into an initially empty tree. Insert in the order given. Show the tree after each insert. *(Source – directly off of the homework)*

\

Continued on next page

5. (5 pts) Write code to print the items of a 2-3 tree using preorder traversal. You may assume that the node member variable *numItems* is set to represent the number of items currently stored in that node and that for nodes with only one item, that item is stored in the *small* member variable. Use the following class definitions:

```
class Tree {
friend class Node;
private:
        Node* root;
        void preorder(Node*);
public:
        void preorder();
};
```

```
class Node {
private:
        Node *left, *middle, *right;
        itemtype small, large;
        int numItems;
};
```

```
void
Tree::preorder ( Node* cur ) {
```

Yea!