# CIS 263

Midterm

## Instructions

1. Questions are short response, not short answer. Use sentences and be clear.
2. Carefully read the questions – some are multi-part.
3. Put your name on this exam – five points for doing so.
4. Write clearly as possible.
5. The test is 100 points total.
6. What is the Big-O time of this algorithm (10 points):

void doStuff(int n){

int x=1;

for(int i=0; i<n; i++){

for(j=0; j<n; j++){

x = x + j;

for(int k=0; k<j; k++){

x = x + k;

}

}

}

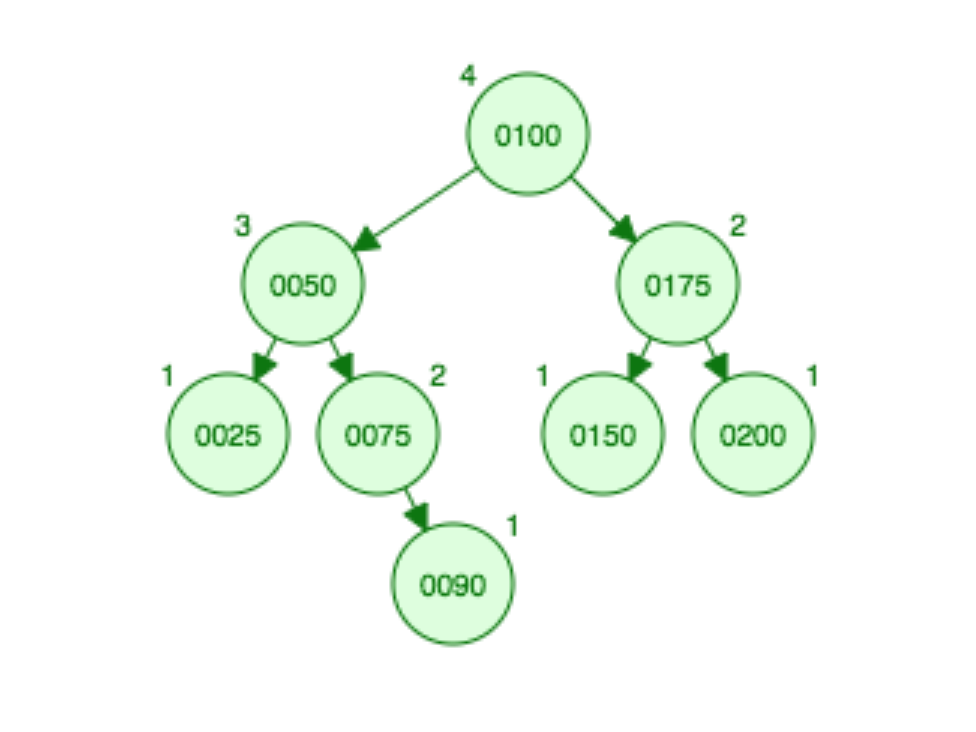
for(int i=0; i<n; i++){

x = x – 1;

}

}

1. Define “Abstract Data Type” (ADT) (5 points).
2. Describe the list ADT (5 points).
3. Describe the two implementations of the list ADT we discussed in class. What are advantages and disadvantages of each? (10 points).
4. What is an iterator? How do we declare and use one in C++? (10 points).
5. Imagine we have a pointer in C++ that is being used as an array. Call this pointer “data”. This data structure currently holds 100 elements. Write code below that will increase the number of elements it will hold to 1000. Don’t use malloc or free. (10 points).
6. What is an adapter class? Why would we use one? (5 points).
7. Define a tree using the mathematical definition we used in class. Draw an example of an expression tree, a generic binary tree, and a binary search tree (15 points).
8. Given the following AVL tree, insert 80 and balance the tree. Show each step. (12 points).



1. We have a BST made out of Nodes. The class structure of Node is

template <typename T>

class BinaryNode {

T element; // Data this BinaryNode holds

Node\* left; // Left child

Node\* right; // Right child

};

void goCrazy( BinaryNode \*t, ostream & out ) const

{

if( t != nullptr )

{

T tmp = t->element;

goCrazy( t->left, out );

if(tmp < t->right){

tmp = null;

}

goCrazy( t->right, out );

out << t->element << endl;

}

}

What type of traversal does this algorithm perform? If the BST holds the data (50, 12, 87, 10, 12, 92, 102) - inserted in that order- what will ultimately be printed out? (13 points).