CSCE4133/5133 – Algorithms
Fall 2024

Quiz 2 - Answers

Date: Sep. 13, 2024

Time: 30 minutes

Instructions:

- Write your full name, email address and student ID in the report.
- Submission via BlackBoard or paper

```
class Simple_tree {
   private:
      int node_value;
      int height;
      Simple_tree *parent_node;
      Single_list<Simple_tree *> children;

   public:
      Simple_tree(int = 0, Simple_tree * = nullptr );

      int value() const;
      Simple_tree *parent() const;
      int degree() const;
      bool is_root() const;
      bool is_leaf() const;
      Simple_tree *child(int n ) const;
      int print_nodes(int h) const;
      int sum() const;
};
```

Figure 1. Sample Tree

Question 1: (50 points)

Write the C++ function to compute the summation of values in the tree.
int sum() const

For example, given the tree in Fig. 1, sum() will return (8+14+16+19+23+27=107).

```
int Simple_tree::sum() const {
   if (this == nullptr) {
      return 0;
   }

   int tree_sum = node_value;

   for (auto *child = children.begin(); child != children.end(); child = child->next) {
      tree_sum += child->sum();
   }

   return tree_sum;
}
```

Question 2: (50 points)

Given a value of h, write the function to print out all subtree roots in the tree with a height equal to h. Assume that the height of each node has been precomputed and assigned in the attribute height.

int print_nodes(int h) const

The height of a subtree is the maximum distance from the subtree root to the leaves.

For example, given the tree in Fig. 1, print_nodes(1) will print out 16.

```
int Simple_tree::print_nodes(int h) {
   if (this == nullptr) {
      return -1; // Can return anything
   }

   for (auto *child = children.begin(); child != children.end(); child = child->next) {
      child->print_nodes(h);
   }

   if (this->height == h) {
      std::cout << node_value << '\n';
   }

   return this->height; // Can return anything
}
```

Question 3: (50 points) (Graduate Students Only)

Suppose the tree is a binary Tree (each node has maximum two children) with the height of 3, what is the minimum and maximum memory (in bytes) required for this tree structure?

Minimum number of nodes: 4
4 * 2 * sizeof(int) + 4 * sizeof(int) + (4 * 5 – 1) * sizeof(void*)
= 4 * 2 * 4 bytes + 4 * 4 bytes + (4 * 5 – 1) * 4 bytes = 124 bytes

Maximum number of nodes: 15
15 * 2 * sizeof(int) + 15 * sizeof(int) + (15 * 5 – 1) * sizeof(void*)
= 15 * 2 * 4 bytes + 15 * 4 bytes + (15 * 5 – 1) * 4 bytes = 476 bytes