CSC 3100: Data Structures

Final Exam (close book)

Time: 08:30am - 10:30am (120 mins), May 19, 2022

1. [10 marks] State and prove whether the following statements are correct or not:

(1) [5 marks] $n^{1/2} + logn = O(n^{1/2})$

(2) [5 marks] $log_{10}(2^n) = \Theta(n)$

2. [10 marks] Suppose the nodes of a doubly linked list structure are defined as follows:

```
Public class Node {
     public int data;
     public Node next, prev;
};
```

Design an algorithm with pseudocodes which **concatenates** two given lists (the first node of the second list will follow the last node of the first list) and returns the new list. Note that it does not create new nodes; it just rearranges the links of some existing nodes.

3. [10 marks] RadixSort (*this problem was selected from lecture notes*):

(1) [7 marks] Given a sequence of integer values: [170, 045, 075, 090, 002, 024, 802, 066], show how to use RadixSort to sort it in the **ascending** order.

(2) [3 marks] In RadixSort, can we start from the **most** significant digit? If yes, explain why; if no, give a counterexample.

4. [10 marks] Given the **inorder sequence** *X* and **postorder sequence** *Y* of traversing a binary tree, reconstruct the binary tree such that its **inorder** and **postorder** are *X* and *Y* respectively. Here, *X* and *Y* are arrays with elements *X[1]*, *X[2]*, …, *X[n]* and *Y[1]*, *Y[2]*, …, *Y[n]* respectively, where $n \geq 1$.

(1) [2 marks] Briefly discuss the main idea of reconstruction.

(2) [4 marks] Write the pseudocodes for the reconstruction algorithm.

(3) [4 marks] Analyze the time complexity of the above algorithm using the Θ notation.

5. [10 marks] Given an array of keys *A*=[4, 8, 6, 9, 11, 1, 12], use **HeapSort** to sort it:

(1) [5 marks] Draw a sequence of figures to show how a **max-heap** is built on *A*.

(2) [5 marks] Draw a sequence of figures to show how the elements are sorted **ascendingly** by HeapSort, such that each figure shows an updated max-heap after deleting a key.

6. [10 marks] Consider a hashtable with separate **chaining** with *N* buckets and *k* items.

(1) [2 marks] *k*/*N* is the definition of a term used when discussing hashing. What is the name of this term?

(2) [2 marks] Is it necessary that $k < N$?

(3) [3 marks] In the worst-case, how many items could be in a single bucket?

(4) [3 marks] If we resize the table to a table of size *2N*, what is the asymptotic running time in terms of *k* and *N* to put all the items in the new table?

7. [10 marks] Given a binary search tree with root node *T*, find which value is the **median** value, and delete that value. Assume that for each node *p*, we can access its left child and right child by *p.leftChild* and *p.rightChild* respectively, and access its key value by *p.key*.

(1) [5 marks] Design an algorithm to solve the above problem and show its main steps.

(2) [5 marks] Use O notation to analyze the time complexity of your algorithm. Your bound should be as tight as possible (as if you are using Θ notation).

8. [10 marks] Consider an undirected graph with *n* nodes and *m* edges ($m \geq n$), and its adjacent list, where for each node *v*, its adjacent list is denoted by *Adj(v)*.

(1) [3 marks] What is the time complexity to count the **total number of edges**? Use O notation and the bound should be very tight (as if you are using Θ notation).

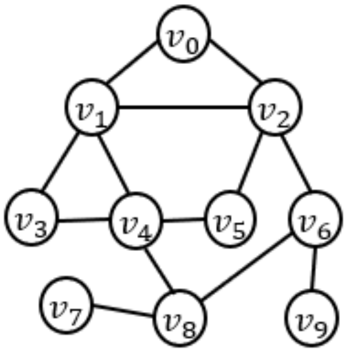
(2) [4 marks] Assume that the size of the adjacent list of each node *v,* denoted by *d(v)*, is known in advance, can we **sort all the nodes according to their degrees** in an ascending order in *O(n)* time cost? If yes, briefly show the idea; if no, please explain why.

(3) [3 marks] What is the big-O time cost of **removing a node *v* from the graph**? Notice that removing a node means that all the edges linked to this node will be removed as well, and the bound should be very tight (as if you are using Θ notation).

9. [10 marks] Consider an undirected graph G_1, which is depicted as below:

(1) [6 marks] Show the sequences of nodes of **BFS traversal** and **DFS traversal** respectively, by assuming that the starting node is set to v_0.

(2) [4 marks] Compare BFS with DFS in terms of time cost and extra space cost, if we use the **adjacent matrix** to represent the graph.



Graph G_1

10. [10 marks] You are given an edge-weighted undirected graph, using the adjacency list representation, together with the list of edges in its minimum spanning tree (MST). Describe an efficient algorithm for updating the MST, when each of the following operations is performed on the graph. Assume that common graph operations (e.g., DFS, BFS, finding a cycle, etc.) are available to you, and don't describe how to re-implement them.

(1) [5 marks] Update the MST when the weight of an edge that **was** part of the MST is **increased**. Show the main ideas of your algorithm and give the order-of-growth running time of your algorithm as a function of *V* and/or *E*.

(2) [5 marks] Update the MST when the weight of an edge that **was not** part of the MST is **decreased**. Show the main ideas of your algorithm and give the order-of-growth running time of your algorithm as a function of *V* and/or *E*.