CS101 Data Structures
Sample Midterm *Question* Book [Fall 2019]

Instructor: Dengji Zhao, Yuyao Zhang

Time: 10:15-12:00, Nov 6th, 2019

Candidate Information

- *Student ID*:
- *Name (Chinese)*:
- *Seat number*:
- *Signature*:

Format: Closed

- You are **not** allowed to bring any papers, books or electronic devices including regular calculators.
- You are **not** allowed to discuss or share anything with others during the exam.
- **Answer the questions in the *Answer Book* only**.

1 Multiple Choice (choose one correct answer)

Question 1 *What is the post-order of the tree?*

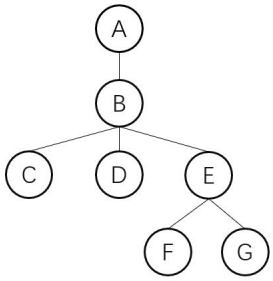


Fig. 1: Find the post-order of the tree

(A) CDFGEBA
(B) ABCDEFG
(C) CDFEGBA
(D) CDBFEGA

Question 2 *Suppose pointer q points to node A, pointer p points to node A.next and pointer t points to node B. If we want to insert node B immediately after node A, which of following operations is in right order?*

(A) s→next = p→next; p→next = s
(B) p→next = s; s→next = q
(C) Both A and B are right
(D) Neither A nor B is right

Question 3 *Which data structure is required for Breadth First Traversal on a tree?*

(A) Stack
(B) Array
(C) Queue
(D) Tree

Question 4 *Which of the following stack operations could result in stack underflow?*

(A) Is_empty
(B) Pop
(C) Push
(D) Two or more of the above answers

Question 5 *The initial configuration of a queue has four integers "1,2,3,4" (1 is in the front). In order to get the configuration of "4,3,2,1", one needs at least*

(A) 3 enqueues and 4 dequeues
(B) 3 enqueues and 2 dequeues

(C) 2 enqueues and 3 dequeues
(D) 3 enqueues and 3 dequeues

Question 6 *What is the worst case complexity of insertion sort?*

(A) $O(n)$
(B) $O(log\ n)$
(C) $O(n\ log\ n)$
(D) $O(n^2)$

2 Multiple Choice (choose ALL correct answers)

Question 7 *Choose the correct statements of tree:*

(A) A non-empty tree of degree with 4 or 0 must contain odd number of nodes.
(B) The sum of the degree of nodes in a tree is equal to the number of nodes - 1.
(C) The height of a tree with degree of 3 and 50 nodes must be larger than 3(the height of a tree with only root node is 0).
(D) If the post-order and pre-order of a tree are known, then the structure of a tree is unique.

Question 8 *Which of the following statements about Prim's algorithm is true?*

(A) Prim's algorithm has lower time complexity than Kruskal's algorithm.
(B) Prim's algorithm cannot work if there are negative weights on edges in the graph.
(C) Prim's algorithm cannot work if there exists negative cycles in the graph.
(D) The statements above are all false.

Question 9 *Select the time complexity for each operation/algorithm. Each option can be chose more than once.*

- *Erase the kth element at the back of the single linked list.*
- *Bubble sort (non-optimal)*
- *Insertion Sort (worst case)*
- *Breadth-First Traversal on a tree*
- *Depth-first Traversal*
- *Heap sort*
- *Pop the top object of a n nodes binary heap*
- *Element insertion to a non-empty perfect binary tree*
- *Breadth-First Traversal on a graph.* $|V| = n$, $|E| = m$, $O(m + n)$.

(A) $O(1)$
(B) $O(n)$
(C) $O(n^2)$
(D) $O(\log n)$
(E) $O(n \log n)$
(F) $O(n^2 \log n)$

(G) $O(m + n)$

Question 10 *Which of the following statements is true?*

(A) For any node in an AVL tree, the left child tree of it is an AVL tree
(B) For any node in a binary heap, the left child tree of it is a binary heap
(C) A subtree of a binary heap is a binary heap
(D) A subtree of a binary search tree is a binary search tree

Question 11 *What is the max difference in the depths between the leaves of a AVL tree with 143 nodes?*

(A) 3
(B) 4
(C) 5
(D) 6

Question 12 *Given a binary tree, we know the pre-order traversal of the tree is 7,5,3,9,6,4,1, and the in-order traversal of the tree is 3,5,9,7,6,4,1, which of the following sequences is the post-order traversal of it?*

(A) 3,9,5,1,4,6,7
(B) 3,1,5,9,6,4,7
(C) 5,1,6,3,9,7,4
(D) 6,1,3,9,7,4,5

Question 13 *Which of the following applications can use disjoint sets?*

(A) Checking an equivalence relation
(B) Maze generation
(C) Checking a cycle in the graph
(D) none of above

Question 14 *In the process of building disjoint sets, you will have several trees after every union. Assume that after one union, there exists k trees, which is $t_1, \cdots, t_k$. Let x_k be the number of nodes whose heights are r in t_k. Let $A = \sum_{i=1}^{k} x_i$. What is the possible value of A? N denotes the total number of nodes.*

(A) N
(B) $\lfloor \frac{N}{2} \rfloor$
(C) $\lfloor \frac{N}{2^r} \rfloor$
(D) 1

Question 15 *Which statement(s) is(are)* ***not*** *true about binary tree?*

A If a binary tree has n leaf nodes, then the number of nodes of degree 2 is $n - 1$.
B In a complete non-empty binary tree with height h ($h \geq 2$), there must exist a perfect sub-tree with height $h - 1$ in this tree.
C Suppose there are M ($M \geq 1$) leaf nodes l_1, l_2, $\cdots$, l_M in a binary tree. The depth of each node is d_1, d_2, $\cdots$, d_M respectively. If $M \to \infty$, then $\lim_{M \to \infty} \left[\max \left(\sum_{i=1}^{M} 2^{-d_i} \right) \right] = 2$.
D None of the above.

3 True or False

Question 16 *In a widely accepted manner, depth first search is often implemented by a queue and the breadth first search is implemented by a stack and recursion.*

Question 17 *Given a binary tree with n nodes, we can check whether it is a binary search tree in* $\Theta(\log n)$ *time.*

Question 18 *When we want to implement a STACK(QUEUE), we can use a linked list which has a head pointer and a tail pointer in order to make sure both Push(Enqueue) and Pop(Dequeue) operations can be realized in* $\theta(1)$ *time.*

Question 19 *We cannot find a topological sort of a directed acyclic graph G in* $O(|E| \log |E|)$ *time.*

Question 20 *The time complexity of building a heap using Floyd's Method is* $\Theta(n \log n)$.

Question 21 *Stacks are important in compilers and operating systemsinsertions and deletions are made only at one end of a stack, its top.*

Queues represent waiting lines; insertions are made at the tail of a queue and deletions are made from the head of a queue.

Question 22 *When we say that an algorithm X is asymptotically more efficient than Y, we mean X will always be a better choice for large inputs.*

Question 23 $2f(n) + O(f(n)) = \Theta(f(n))$

Question 24 *Let v be the only node of a Max Heap that does not satisfy the Max Heap Property. Running the* `MAX-HEAPIFY()` *procedure on v will always result in a Max Heap for which all nodes satisfy the Max Heap Property.*

Question 25 *Suppose there are* M $(M \geq 1)$ *leaf nodes* $l_1, l_2, \cdots, l_M$ *in a binary tree. The depth of each node is* $d_1, d_2, \cdots, d_M$ *respectively. If* $M \to \infty$, *then* $\lim_{M\to\infty} \left[\max\left(\sum_{i=1}^{M} 2^{-d_i}\right)\right] = 2$.

4 General Questions

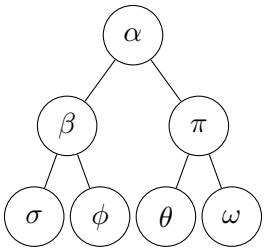
Question 26 *We wish to determine the most efficient way to deliver power to a network of cities. Initially, we have n cities that are all disconnected. It costs* $c_i \geq 0$ *to open up a power plant at city i and* $r_{i,j} \geq 0$ *to build a cable between cities i and j. A city is said to have power if either it has a power plant, or it is connected by a series of cables to some other city with a power plant. Give an efficient algorithm for finding the minimum cost to power all the cities. (Hint: try to take advantage of minimum spanning trees.)*

Question 27 *Running time* $T(n)$ *of processing n data items with a given algorithm is described by the recurrence:*

$$T(n) = k * T(\frac{n}{k}) + c * n$$
$$T(1) = 0$$

Derive a closed form formula for $T(n)$ *in terms of c, n, and k. What is the computational complexity of this algorithm in a Big-O sense? Hint: To have the well-defined recurrence, assume that* $n = k^m$ *with the integer* $m = log_k n$ *and k.*

Question 28 *Consider the tree on the left where Greek letters represent numerical values. In the boxes of the tables, shade all values that might match the text. Assume all values are unique. For BSTs, assume left items are less than. When treating the tree like a graph, assume nothing about the order of adjacency lists.*



MinHeap, largest item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
MinHeap, smallest item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
BST, largest item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
BST, smallest item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
MinHeap, median item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
BST, median item	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
MinHeap, new root after Pop	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	
MinHeap, root after inserting new item γ	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	☐ γ
BST, root after inserting new item γ	☐ α	☐ β	☐ π	☐ σ	☐ ϕ	☐ θ	☐ ω	☐ γ

Question 29 *Prove by induction that the number of inversions in an array of size n is* $\leqslant \frac{n(n-1)}{2}$*. The head of proof has been given to you(3pt). Point out when the equity establishes (what the array is alike when there is* $\frac{n(n-1)}{2}$ *inversions in an array of size n)(2pt).*

Solution:

1. *If n=1, we have only one permutation which has no inversions.*
2. *Then, suppose ....*

(Compensate your proof here)