Data Structure: Review Topics and Questions

Simple abstract data types
Linked list (Double and single), ArrayList, Linear lists, stacks, queues: concepts, implementation details, and applications
Expect three types of questions:

- Knowledge: for example, define what stacks/queues…. are, or how does (X) work?
- Coding: for example, write Java code for (X) (like the examples given in the slides).
- Problem/application: for example, Tracing or dry-run any of these algorithms (like the examples in the slides)

Trees
Binary and other trees
Priority queues
Expect three types of questions:

- Knowledge: for example, define what binary tree/priority queues …. are, or how does (X) work?
- Coding: for example, write Java code for (post-order binary tree traversal X) (like the examples given in the slides).
- Problem/application: for example, solve problems, for example, heapify() or adjust(), expression trees, binary tree (like examples we did in the tutorials).

Graphs
Expect three types of questions:

- Knowledge: for example, define what graphs (adjacency matrices, breadth-first search) are.
- Coding: for example, write Java code for (X: breadth-first search) (like the examples given in the slides).
- Problem/applications: for example, undirected/directed graph, breadth-first search tree, depth-first search, draw a graph or get set of vertices of a graph (like examples we did in the tutorials), also giving graphs and asked to calculate the minimum cost tree using any of these algorithms (like the examples in the slides)

Hashing
Expect three types of questions:

- Knowledge: for example, define what hashing, hash collision, linear probing and chaining are.
- Problem: for example, dealing with overflows using linear probing technique (like examples we did in the tutorials).

Any other topics would be covered by the same style of questions

Examples of questions:

1. Choosing the proper data structure depends on the application. Specify what data structure you would choose in each of the following cases. You can choose from a static array, singly linked list, doubly linked list, an array linear list, a queue, a heap, or a stack.
    a) An application requires a structure where new nodes can easily added to the front and removed from the back **(1 Mark)**
    b) An application requires a data structure that can have priority for processing **(1 Mark)**
    c) An application requires processing the last item arrived to the system. **(1 Mark)**

2. Dijkstra's two-stack algorithm is used for the evaluation of fully parenthesized arithmetic expression and it works as follows:

    - Operand (value): push onto the operand (value) stack
    - Operator: push onto the operator stack
    - Left parenthesis: ignore
    - Right parenthesis: pop operator and two values; push the result of applying that operator to those values onto the operand stack

Follow the above algorithm and evaluate the following expression: **(5+ ((2 + 3) * (4 * 1))). Show your evaluation step by step** **(4 marks)**

3. Is the array with values **{23,17,14,6,13,10,1,5,7,12}** a **max-heap**? If not, why not? Draw the max-heap tree of this array to support your answer. **(3 marks)**

4. Is the array with values **{10,20,100,25,30}** a **min-heap**? If not, why not? Draw the max-heap tree of this array to support your answer. **(3 marks)**
5. Illustrate the operation of Max-Heapify(A,3) on the array **A = {27,17,3,16,13,10,1,5,7,12,4,8,9,0}**. Is the result a max-heap? If not, why not? Hint: Max-Heapify is a method to correct any problem in the heap tree such that making the tree a max heap. **(6 marks)**
6. Given the below combination of tree traversal sequences, construct the binary tree from which these sequences came: (7 marks each)
    (a)
    in-order: g d h b e i a f j c
    post-order: g h d i e b j f c a
    (b)
    in-order: g d h b e i a f j c
    level order: a b c d e f g h i
    (hint: section 08 shows a detailed example)
7. Giving a hash function, h(x)=x mod 13, insert the following values into a hash table using linear probing to avoid hash collision. The values are 18, 41, 22, 44, 59, 32, 31, 73

8. Given the weighted directed graph below, construct its cost and adjacent matrix. (4 marks)