Rubrics

Quiz I (TOTAL OF 20 POINTS), Jan 14, 2020
CSE 102: Data Structures and Algorithms

Note: Since it's a theoretical quiz, students might have written very different answers. So, give appropriate marks based on certain keywords in the answers. Do partial marking.

1. Mention at least three reasons for using data structures? [3 points]

i. Computational efficient ways of analyzing, storing, searching and modeling data [1]
ii. provide a means to manage large amounts of data efficiently [1]
iii. efficient data structures are key to designing efficient algorithms [1]

2. What is an abstract data type? Why are they necessary? [3 points]

ADT is an abstract model of a data structure together with the operations (can be treated as a form of algorithms) processed on the data structure. An ADT does not specify how the data type is implemented. **Example:** Stack, Queue etc. **[1-Definition, 0.5-Example]**

OR

Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of value and a set of operations.
The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented. **Example:** Stack, Queue etc.

They are necessary because: **[1.5]**

- they give an implementation independent view.
- a user only needs to know what a data type can do but not how it will do it.
- ~~We can think of ADT as a black box which hides the inner structure and design of the data type.~~

(Any other valid reason)

3. What are recursions? How are they different from Iterative solutions? [3 points]

Definition: Recursion is the process of solving a problem by dividing it into smaller sub-problems of the same form. In recursion, a function calls itself repeatedly until some base condition is not reached. (This means that recursive programs will use the same function or the method to solve sub-problems at different levels of the solution.) **[1.5]**

Differences: **[1.5]**

- In iteration, some statements are executed again and again using loops until the controlling condition becomes false.
- Recursive solutions are slower than corresponding iterative solutions.
- Recursive solutions are often simpler and more elegant than iterative solutions

4. What are the steps to generate a recursive solution? [2 points]

1. Identify simple cases that can be solved without recursion. (base case) **[1]**
2. Find a recursive decomposition that breaks each instance of the problem into simpler sub-problems of the same type, which can be solved by applying the method recursively. (continuation case) **[1]**

5. Why do recursive solutions generally have higher computational overhead? [2 points]

- Recursion has the overhead of multiple method invocations
- In recursion, extra effort is required to push data onto a stack and later pop it off.
- Recursion takes a lot of stack space, as each time a method calls itself, a pointer to it and its local variables are generated again.
  (Any 2 points)

6. Why is there emphasis on designing efficient algorithms in the era of decreasing computational costs? [3 points]

In today's scenario, the amount of data is increasing exponentially. It has almost doubled in the last three years.
Inefficient algorithms, even on the fastest PCs may take years to process.
Also, if the input size is very large, space complexity also increases.
So, we require efficient algorithms with low time and space complexity. **[3]**

7. How do you measure running time of an algorithm? Justify your answer. [4 points]

The running time of an algorithm for a specific input can be measured by counting the number of operations executed or by counting no. of instructions executed by algorithm in RAM model.
The greater the number of operations, the longer the running time of an algorithm. **[3]**

Justification: Example to justify **[1]**