

Major topics that I expect you to know

1. Insertion, deletion, search in sorted and unsorted array. Complexity.
2. Abstract classes and interfaces. Differences.
3. What is ADT? Is interface an ADT? Why do we need interface? How to implement it?
4. JUnit testing. What is the purpose? How to write a simple a test case.
5. Singly and doubly linked lists. You should be able to
 - a. add a new node at any position (list with head only, or head and tail)
 - b. remove a node from any position (list with head only, or head and tail)
 - c. search for a given node.
 - d. Iterate through the list.

What is the running time for these operations?

6. Iterators. Implementation, purpose. What is the difference between Iterator and ListIterator.
7. Generics. Why do we need it? What problem does it solve? How to make a class generic? How to create a generic object? Diamond operator.
8. Complexity: definition of big-O, big-Theta, big-Omega. Determine the running time for a given piece of code.
9. Stack and Queues. Different Implementations. Running time for pop, push, enqueue and dequeue. Adapter design pattern: what is the idea, can we use it to implement a stack? How? Queue? Does it matter what end of the linked list (singly, doubly) will we choose? How about ArrayList, can it be used for stack implementation?
10. Recursion. Why do we need base cases? What happens if you do not make the problem smaller (see you in Recursion Land ☺).
11. 6 types of sorts. You should be able to trace each one given unsorted array. Know and understand the implementation of at least insertion, merge and quick sorts. Running times for best, average and worst cases for all 6 algorithms.
12. Heaps: tree and array representation, max and min heaps. How to build a heap, remove from a heap. Priority Queue using heaps. D-heaps (advantage/disadvantage)
13. Binary trees, tree traversals. BFS, DFS. Binary search trees, how to insert, delete, search.
14. Hash tables: collisions and ways to resolve them. Load factor, complexity. Hash function.
15. Intro to C: definition of pointers, structures, copy by values, string. Be able to understand the code from the slides and explain it.
16. Logarithm properties.
17. Inner classes: properties.

You need review your doubly linked list and iterator implementation.
You must know all the running times for all algorithms we talk in class.

Nice link: <https://quizlet.com/28620556/cs-314-novak-flash-cards/>