Exam, Data Structures

Solve as many of the following problems as you can. All questions are weighed equally. Be concise and to the point (irrelevant answers will not help towards partial credit). As a general rule, you should not spend (assuming you know the answer perfectly) more than five minutes on any single question.

The time allotted for the exam is two hours. The exam is **closed book**.

Cheating attempts lead to failing the course.

1 Exam questions

1. Describe the evolution of Selection sort on the following set of integers: 5 2 6 4 1 3.
2. Write the interface (header file) for the classes implementing a *doubly linked* list.
3. Write a function (or set of functions) to insert a node *at the front* of a linked list.
4. Write a small program illustrating the use of list in the STL.
5. What is the advantage of using a skip list in your program, compared to a simply linked list ? Give one disadvantage you can think of.
6. Write the Binary Search Tree property. Show (by writing a small function) how do we use this property to locate an element in a binary search tree.
7. List the sequence of vertices obtained by a *postorder traveral* of a complete binary tree with seven nodes, labeled from 1 to 7 from top to bottom, left to right.
8. What are the operations supported by the MAX-HEAP data structure ?
9. Define the red-black properties. What is the advantage of using red-black trees, compared to an ordinary BST ?
10. How do we solve collisions in hashing using chaining ? What auxiliary data structure do we use ?
11. What operations does the Interval-Tree data structure support ?
12. Describe the idea of the algorithm for computing the closest points (**Only the idea, not the details/justification**).
13. Draw an example (and write the fragment of code) necessary to implement a RIGHT-ROTATE operation (used e.g. in red-black trees).
14. What data structure we learnt in this course is used to implement iterative versions of recursive functions ? Name one such nonrecursive implementation we saw in this course (just name it, do not write the pseudocode).
15. Present one possible application of some of the data structures (and algorithms) you learned about in this course.