

## CSE 330 Final – Topics of Test Problems

### Binary Search:

- Know the algorithm, review its implementation in C++ (e.g., textbook, or sample solution under Lab6)
- Know important constraints under which this algorithm will work
- Know its performance in numbers of operations relative to the number  $N$  of elements stored
- Know the data structure(s) this algorithm is operating on best (we have seen it with vectors, can you imagine other data structures? ... think in terms of efficiency of performance).

### Binary Search Trees:

- What makes a binary tree a binary search tree? What is the “binary search tree property”?
- Be able to identify a binary search tree by its graph
- Know the process by which a new element is inserted into a BST
- Know the process by which an element is found in BST
- Know the process by which an element is removed from a BST
- Know the algorithmic complexity,  $O(?)$ , for these operations; are they best case, worst case, average case
- Know different systematic ways to traverse (visit each node) the binary search tree (left-root-right, root-left-right, etc.) ... which one seems to be good match for traversing binary search trees

### Sets:

- Understand how class Set benefits from the implementation of class BinarySearchTree
- Know how to \*use\* Set iterators (no need to understand the particular details of iterator I defined); that is declare, initialize, apply \* and ++ operators; know how to set an iterator to the beginning of a Set<T> and how to test for the end of the Set<T>
- Understand how/why the Set<T> data structure in the way we defined matches up well with the mathematical definition of sets.

### Maps:

- Know what nature and purpose of a Map data type; what is it good for? In what way can it be seen as a generalization of the vector data type?
- Understand the clever trick that we used to define the Map data structure in terms of a Set data structure
- Understand how the computational costs for Set operations apply to the computational costs of Map operations.

- Know the Map operations and have an understanding of why they are fewer than the Set operations; e.g., we do not have Map iterators, ... why not?

### Priority Queues/Heaps

... we did not spend a lot of time on these; however, know the basic concepts ...

- Know what a “complete” binary tree is
- Know what a “heap” is; i.e. know the “heap property” ...
- Know how a complete binary tree can be represented in a vector;
- Know how to find the left and right children, and the parent of a node when the complete binary tree is given as a vector
- Know how to do graphically, on paper, the operations of inserting a value, and popping the root value off the priority queue so that the resulting data structure retains its heap property.

### General Understanding:

Know the pros and cons of each data structure. Have an understanding of the types of applications for which each data structure would be most suitable (and for which they are not).

### Coding:

You will not be asked to write extensive, many-line C++ code, but you should be able to read and understand the kind of C++ code that we have been dealing with in the context of our labs and homeworks. Be able to provide code fragments in a given context. Clearly, CSE 330 without a reasonably good understanding of C++ is unthinkable, so be ready to apply your skills.