

Homework 2

Search Trees and Hash Tables

Due: Monday, November 13, beginning of class

This assignment is designed to give you practice with various implementations of the search and dictionary ADTs we have seen.

Note: This is an individual assignment. (Please review the collaboration policy on the course web page. You **must** mention the names of fellow students if you work with someone on this assignment.)

Total: 72 points.

1. **[Search Trees]** (20 points) Show the result of inserting 15, 20, 17, 8, 6, 18, 9, 11, 23, 19 into the following initially empty search tree data structures. It's fine to turn in only the final result. However, if your final answer is incorrect, intermediate steps may allow partial credit.
 - a. Binary search tree (unbalanced)
 - b. AVL tree
 - c. B-tree with $M=3$, $L=2$
 - d. B-tree with $M=3$, $L=3$
2. **[Hash Tables]** (20 points) Do the same as you did in the above problem, but now for the following hash table implementations. Indicate if and when no more keys can be inserted into the table. Use $h(x) = x \bmod \text{tableSize}$.
 - a. Hash table of size 11 with separate chaining where each bucket is an unsorted linked list
 - b. Hash table of size 11 with open addressing and linear probing
 - c. Hash table of size 11 with open addressing and quadratic probing
 - d. Hash table of size 11 with open addressing and double hashing, where $\text{hash}_2(x) = 5 - (x \bmod 5)$
 - e. Hash table of size 11 with open addressing, quadratic probing and rehashing when the table is half full
3. **[Range Search]** (20 points)
 - a. Give an algorithm (pseudocode) that takes a binary search tree as input along with two keys x and y , with x less than or equal to y , and prints out all keys z in the tree that lie between x and y inclusive.

- b. What is the worst-case running time of your algorithm given in part a? Give a bound in terms of the tree size N , the tree height H , and the number M of keys output.
 - c. Give an algorithm (pseudocode) that instead of a binary search tree, uses a hash table of size `tableSize` with separate chaining and N elements where each bucket is an unsorted linked list.
 - d. What would be the worst-case running time of your algorithm given in part c. Give a bound in terms of `tableSize` and N .
4. **[Open Addressing]** (12 points) Problem 5.8, page 195.