

Computer Science 241

In-Class Exercises

1 Hashing

1. Properties of hash functions

(a) What properties must a function satisfy to be a valid hash function?

(b) What properties would we like a hash function have in order to be well-behaved when used for storing keys in a hash table?

2. Suppose the keys in your hash table are integers, and that they are random and uniformly distributed (i.e., any integer key is as likely as any other). Consider the following candidate hash functions. You may assume the following:

- In all cases, you will compute the final bucket address of a key k as $h(x) \% C$ where C is your table size.
- Assume you have written a method `randint` that generates a random integer.
- Your table size C is prime.

a. $h(x) = 0$

b. $h(x) = x$

c. $h(x) = 2x$

d. $h(x) = x^2$

e. $h(x) = x^x$

f. $h(x) = x \% 10$

g. $h(x) = x * \text{randint}()$

(a) List which of the above functions are *valid* hash functions:

(b) List which of the above functions are *good* hash functions:

(c) Suppose your table size C is even. Does your answer to the previous part change? If so, give a new list which hash functions are good:

2 Word problems!

For each of the problems below, write an **English** description of how you'd solve the problem. Specify any data structures used, and feel free to rely on algorithms, operations, etc. covered in this class. If you finish describing your approach to all of the problems, write out pseudocode for your solution.

0. **Example Problem:** Given a directed graph, design an algorithm to find out whether there is a route between two nodes.

Example Solution: Given a source and a destination node, run a depth-first search starting from the first one. Each time a node is visited, check whether it matches the destination node and return true if so. Return false if the DFS finishes.

1. Given a sorted (increasing order) array with unique integer elements, write an algorithm to create a binary search tree with the minimum possible height.

2. You have a text file with 1 million URLs. How would you detect duplicate URLs?

3. (1 pt) Implement a function to check if a given binary tree is a valid binary search tree.