Advanced computer programming

Exercise session 5: Dictionary

Jean-Michel Begon – http://www.montefiore.ulg.ac.be/~jmbegon

November 2014

Exercise 1

Let `T` be an initially empty binary search tree (BST).

(a) Draw `T` after the insertion of the following keys: 30, 40, 24, 58, 48, 26, 11, 13, 35 and 36 (in that order).

(b) Draw `T` after the suppression of following keys: 13, 58, and 30 (in that order).

Exercise 2

Write an algorithm which computes the number of distinct BST one can build with the set $1, .., N$.

Exercise 3

If one insert the element "z" from the BST `T` (with `TreeInsert(T, z)`) and then removes it (with `TreeDelete(T, z)`), is `T` identical to the original BST ?

Exercise 4

Can removing a key in a BST increase the tree lenght ?

Exercise 5

Let `D` be a 11-slot, initially empty hash table with chaining collision policy. Illustrate the state of `D` with respect to the insertion of the keys

$$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}.$$

The hash function is

$$h(k) = (2k + 5) \mod 11.$$

Exercise 6

Let `D` be a 11-slot, initially empty hash table with a linear probing collision policy $h(k, i) = (h'(k) - i) \mod 11$ (where $h'(k) = (2k + 5) \mod 11$). Illustrate the state of `D` with respect to the insertion of the keys

$$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}.$$

Exercise 7

Let `D` be a 4-slot, initially empty, extensible hash table with a linear probing collision policy $h(k,i) = (h'(k) - i) \mod m$ (where m is the current capacity) whose maximum load factor is $\alpha = 0.75$ (when the load exceed 75%, we copy address-wise the content of `D` in a new hash table with doubled capacity). Illustrate the state of `D` with respect to the insertion of the keys

$$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}.$$

What is the mean access time to an element (the mean time over the given keys of a positive search) ?

Exercise 8

In Java's JDK1.1, the hash function for a String was implemented in the following fashion. What is wrong with this function ?

```
public int hashCode() {
    int hash = 0;
    int skip = length() / 8;
    for (int i = 0; i < length(); i = i + skip) {
        hash = (hash * 37) + charAt(i);
    }
    return hash;
}
```

Exercise 9

Java's hash function for a String is now implemented as :

```
public int hashCode() {
    int hash = 0;
    for (int i = 0; i < length(); i++) {
        hash = (hash * 31) + charAt(i);
    }
    return hash;
}
```

Find two strings of length N which produce the same hash.

Exercise 10

What is the complexity of finding the smallest element in

- a binary search tree;
- a balanced binary search tree;
- a hash table with open addressing;
- a hash table with chaining.

Write a procedure `printLowerThan(T, x)`, which prints the value of `T` which are lower or equal to x. What is the complexity of such an operation ?

Bonus

Bonus 1

Find two constants c_1, c_2 for a quadratic probe with $m = 11$. That is, such that:

$$h(i) = c_1 * i + c_2 * i^2$$

be a permutation of the first 11 integers.

Bonus 2

One would like to represent sparse matrices (*i.e.* containing many zeros) with hash table(s).

(a) What would the key correspond to ?

(b) Give an efficient implementation (data structure, insertion, row/column search, access).

(c) What is the complexity of the insertion and access operations ?

(d) Give an algorithm for sparse matrix multiplication.

(e) Can we do better than with hash tables ?