© Copyright 2024 Saad Mneimneh
It's illegal to upload this document on any third party website

CSCI 705 Algorithms
Homework 6
Due 3/29/2024

Saad Mneimneh
Computer Science
Hunter College of CUNY

Readings
Based on Hashing, Binary Search Trees, Red-Black Trees, and Augmenting RB trees.

Problem 1
Consider a two dimensional array A of integers of size $m \times n$ (and assume $m \leq n$). We say that the array has the criss-cross property if there exists i, j, a, b such that

$$A[i][j] + A[i+a][j+b] = A[i+a][j] + A[i][j+b]$$

For example, the following array has the criss-cross property because

$$A[0][1] + A[2][3] = A[2][1] + A[0][3]$$

1	$\underline{2}$	3	$\underline{4}$	5
5	4	3	2	1
5	$\underline{1}$	2	$\underline{3}$	4
1	5	3	4	2

To find whether an array has the criss-cross property, one can check every pair of rows and every pair of columns, for an $O(m^2n^2)$ time. Design an algorithm that finds whether an array has the criss-cross property in $O(m^2n)$ expected time. *Hint*: The term "expected" should suggest which data structure?

Saad Mneimneh (c) Copyright Material - Illegal To Post

Problem 2: This is experimental
In the past, I have written some code to test the C++ redblack tree and hash table implementations as part of its STL. I will share the code with you here:

`www.cs.hunter.cuny.edu/~saad/courses/alg/hw/redblack_vs_hash.c`

`www.cs.hunter.cuny.edu/~saad/courses/alg/hw/romeo.txt`

The file `redblack_vs_hash.c` has some code that illustrates the use of these structures. The objects are string/int pairs. The code uses the file `romeo.txt`, which contains text from Shakespeare's Romeo and Juliet.

(a) Download the files, compile, and run the code.

(b) Read the comments and make yourself familiar with the use of the STL `map` and `unordered_map` (redblack tree and hashtable, respectively). In the main function, you can switch between the use o f a hash table and a redblack tree. Try both and observe the running time of the program. Is it consistent with what you know about the complexity of operations on redblack trees and hash tables?

(c) Explain what the program is doing (to make sure you understand what's going on). *Hint*: it's about counting words.

(d) You may choose to use a redblack tree or a hash table. Modify the program to do the following:

- Traverse the data structure using its iterator (an example is provided) and add all the pairs to a vector
- Look at the documentation for the sort function in C++ and how it can be used to sort vectors, make the appropriate `#include` and `using` statements to be able to use it. For instance, you might want to look at: `https://cplusplus.com/reference/algorithm/sort/`.
- Provide a comparator function, a function that takes two string/int pairs and returns which one should come first based on the value of the integer.
- Use the sort function on the vector with the comparator function to sort the vector and obtain the 20 most frequent words in `romeo.txt`.

Saad Mneimneh (c) Copyright Material - Illegal To Post

Problem 3
Assume that, in a binary search tree, we augment the information in a node to include the sum of all its predecessors keys plus its own. Consider the following pseudocode to compute `predsum` for each node x in the tree.

```
x = T.minimum()
while x! = T.NIL
   do x.predsum = 0
      y = x
      while y! = T.NIL
         x.predsum = x.predsum+y.key
         y = T.predecessor(y)
   x = T.successor(x)
```

(a) What is the running time of this algorithm?

(b) Suggest a faster algorithm to compute all predsums. Provide a pseudocode similar to the one above.

(c) Describe a recursive algorithm (with pseudocode) that will compute for each node x the sum of all keys in it's subtree (including itself). Call this sum `treesum`. Your algorithm should have linear time complexity.

(d) Which of the two `predsum` or `treesum` is a better information to augment the tree with? Explain.

Problem 4
In many settings, especially when making simulations, one needs to generate items based on a given set of weights. Assume we have a structure that supports the following:

- Add(x): Adds element x with key $key[x]$ and weight $weight[x]$.
- Remove(x): Deletes x from the structure.
- Modify(k, w): Changes $weight[x]$ to w where $key[x] = k$ ($key[x]$ is not affected).
- Generate: returns k, such that $k = key[x]$ with probability proportional to $weight[x]$, i.e. with probability $\frac{weight[x]}{\sum_y weight[y]}$.

We assume all keys are different; for instance, they could be distinct integers. Design a structure that will perform all of the above in $O(\log n)$ time, where n is the number of elements. In particular, attention should be given to the Modify and Generate functions. You can assume the existence of a function $Rand(b)$ that generates a number r chosen uniformly at random from the continuous interval $[0, b]$.

Saad Mneimneh (c) Copyright Material - Illegal To Post

Problem 5

Describe how you will augment a redblack tree to support the operation `even_successor`(x), which finds the node y in the tree with the smallest key k such that $k > x$.key and k is even. We assume that all keys are distinct. The even successor operation should run in $O(\log n)$ time, where n is the number of keys stored in the tree. Describe the operation clearly in English and support it with pseudocode.

Saad Mneimneh (c) Copyright Material - Illegal To Post