University of Waterloo
CS240R Fall 2017
Review Problems

Reminder: Final on Tuesday, December 12 2017

Note: This is a sample of problems designed to help prepare for the final exam. These problems do *not* encompass the entire coverage of the final exam, and should not be used as a reference for what the final exam contains.

True/False

For each statement below, write true or false. Justify three of them.

a) Open addressing hashing that uses linear probing will require two hash functions.

b) Run length encoding may result in text expansion on some strings.

c) When doing range search on a quad tree, if there is no point within the range specified, the worst case runtime complexity is $\Theta(h)$.

d) Suffix trees for pattern matching require preprocessing the pattern.

e) If the root of a 2-3 tree stores only a single key, deleting this key always decreases the height of the tree by 1.

f) If the bubble-up version of *heapify* is used in Heapsort, then the worst-case runtime of Heapsort will be $\Omega(n^2)$.

g) The runtime complexity of range query for kd-trees depends on the spread factor of points.

h) Rehashing may be required in Cuckoo Hashing even if the load factor is at an acceptable value.

i) If an AVL tree node has balance 2 and its right child has balance 1, then a double left rotation is required.

j) Move-to-front compression uses adaptive instead of fixed dictionaries

Multiple Choice

Pick the best answer for each question.

1. The last occurrence function for the pattern `MELSMEMES` would contain the following values for each character:

   a) `E` $= 8$, `L` $= 3$, `M` $= 7$, `S` $= 9$

   b) `E` $= 3$, `L` $= 1$, `M` $= 3$, `S` $= 2$

   c) `E` $= 1$, `L` $= 6$, `M` $= 2$, `S` $= 0$

   d) `E` $= 7$, `L` $= 2$, `M` $= 6$, `S` $= 8$

2. The smallest 2-3 Tree of height 2 contains _____ nodes.

   a) 5

   b) 6

   c) 7

   d) 8

3. Using LZW decoding, the last code 132 decodes to what?

$$67 - 128 - 129 - 130 - 131 - 132$$

   a) CCCCCC

   b) CCCCCCC

   c) CCCCCCCC

   d) CCCCCCCCC

4. A quadtree with bounding box $[0,8] \times [0,8]$ over the following points has a height of _____.

$$(6,2),(0,1),(3,4),(7,5),(1,0)$$

   a) 2

   b) 3

   c) 4

   d) 5

5. Suppose we have an array of n numbers where each number is no larger than n^3, and assume that n is a perfect square. Consider running HeapSort, QuickSort, and RadixSort with radix base $R = \sqrt{n}$ on this array. The worst-case asymptotic runtimes for each sorting algorithm, from best to worst, is:

   a) HeapSort, QuickSort, RadixSort
   b) RadixSort, HeapSort, QuickSort
   c) QuickSort, RadixSort, HeapSort
   d) RadixSort, QuickSort, HeapSort

6. Which one of the following statements about compressed tries is false?

   a) Every internal node stores an index indicating the bit position to be tested on a search.
   b) The root of the compressed trie always tests the first bit.
   c) A compressed trie that stores n keys always contains less than n internal nodes.
   d) The height of a compressed trie never exceeds the length of the longest string it stores.

7. If a length-m pattern does not appear in a length-n text, the following string-matching algorithms still need to read at least $n - m$ characters of the text in the best-case, except for:

   a) DFA
   b) KMP
   c) Boyer-Moore
   d) Rabin-Karp

8. CS240 is a course about

   a) Data structures and algorithms
   b) Unreasonable time management
   c) Reconsidering academic choices
   d) All of the above

Hashing

Using double hashing with the hash functions $h_1(n) = n \bmod 7$ and $h_2(n) = (3n \bmod 6) + 1$, and a table of size 7, answer the questions below:

a) Fill the table with correctly hashed values such that a call to $search(6)$ succeeds at the end of a probe sequence of length four.

b) Suppose the numbers written in your table above were inserted using linear probing instead with the hash function $h_2(n)$. Show the resulting table.

Huffman Compression

a) The following message was compressed using Huffman encoding and transmitted together with its dictionary:

00100000111010101110001011010010

$$
\begin{aligned}
\text{` '} &= 100 \text{ (blank space)} \\
: &= 1011 \text{ (colon)} \\
d &= 1010 \\
\ell &= 010 \\
p &= 001 \\
s &= 000 \\
u &= 11 \\
w &= 011
\end{aligned}
$$

Decompress the string using the dictionary and write the final message.

b) Agent Bond doesn't know the password beforehand, but upon seeing the decoded string, she immediately realizes that the message has been tampered with. Explain how Jane determined this.

Rabin-Karp

For Rabin-Karp pattern matching, consider the following hash function for strings over the alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$:

$$
\begin{aligned}
h(P) = &(\text{\# of occurrences of } \mathtt{A}) + 2 \times (\text{\# of occurrences of } \mathtt{C}) \\
&+ 3 \times (\text{\# of occurrences of } \mathtt{G}) + 4 \times (\text{\# of occurrences of } \mathtt{T})
\end{aligned}
$$

a) Given the pattern $P = \mathtt{TAGCAT}$ and sequence $T = \mathtt{TGCCGATGTAGCTAGCAT}$, use the table below to show all the character comparisons performed during Rabin-Karp pattern matching. Start a new pattern shift (in which character comparison occurs) in a new row. You may not need all the available space.

T	G	C	C	G	A	T	G	T	A	G	C	T	A	G	C	A	T

Table 1: Table for Rabin-Karp problem.

b) Suppose we are searching for a length-m pattern P in a length-n sequence T. After computing the hash value of $T[i \dots i + m - 1]$ (the length-m substring beginning at index i) for $i < n - m$, explain how to compute the hash value of the next substring $T[i + 1 \dots i + m]$ in $O(1)$ time.

c) Give a length-m pattern P and a length-n sequence T, with $n > m$, such that the Rabin-Karp algorithm achieves the worst-case number of character comparisons.

KD-Trees

Consider the following set of points:

$$(80, 3, 44), (52, 70, 8), (70, 96, 12), (94, 20, 15), (65, 98, 54),$$
$$(41, 26, 58), (28, 84, 91), (63, 32, 99), (36, 87, 72), (39, 90, 40).$$

a) Draw the kd-tree corresponding to these points.

b) Draw the subset of the tree that is visited during a range query in the rectangular box $[60, 70] \times [90, 100] \times [50, 90]$.

Range Trees

Consider the x-BST of a range tree shown below:

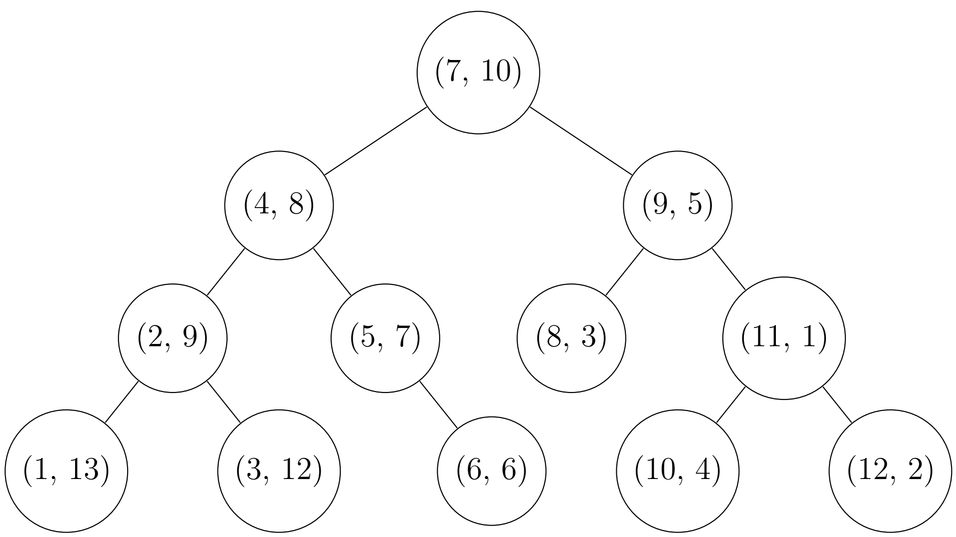

Figure 1: Range Tree x-BST

a) Draw the y-BSTs at nodes $(2, 9)$, $(5, 7)$, and $(9, 5)$.

b) For the query range $R = [0, 7.5] \times [9, 14]$, identify the boundary nodes, inside nodes, and outside nodes for just the x-dimension.

Order Notation

Professor Thick has recently invented a new class of functions called $Onion(f)$. A function $g(n)$ is in $Onion(f(n))$ if there exists a constant $c > 0$ such that $g(n) \leq cf(n)$ for all $n \geq 0$. We assume that $f(n)$ and $g(n)$ are functions that map positive integers to non-negative reals.

a) Give functions g and f such that $g(n) \in Onion(f(n))$.

b) Professor Thick says: "If $g(n) \in O(f(n))$, then $g(n) \in Onion(f(n))$ because O is the first letter of *Onion*". Prove this claim by first principles, or disprove with a counterexample.

Run-Length Encoding

a) Give a string of n bits that achieves the best compression ratio with Run-Length Encoding from all n-bit strings, and state the exact compression ratio achieved.

b) Same question, but for the worst compression ratio. You may assume that n is divisible by 4.

Tries

Given a compressed trie T that stores a list of binary strings, write an algorithm $Consecutive(b_1, b_2)$ that takes two binary strings in T as input, and outputs true if the strings are consecutive in an in-order traversal of the trie, and outputs false otherwise. The runtime should be bounded by $O(|b_1|+|b_2|)$.

For example, suppose T stores $\{000, 01, 0110, 101, 11\}$.
$Consecutive(0110, 101)$ outputs true.
$Consecutive(01, 000)$ outputs true.
$Consecutive(11, 000)$ outputs false.

Lempel-Ziv-Welch Encoding

Encode the following string using LZW compression:

`DARK_DAN_BARKS_DANK`

Add new entries to the encoding dictionary starting at value 128.

Char	ASCII value
A	65
B	66
D	68
K	75
N	78
R	82
S	83
_	95

String Matching Automata

Dr. Taro invented a new string matching automata called NieR: Automata. His robot assistant Pascal discovers that it accepts three patterns:

`2B9S`
`A2`
`B9A`

The alphabet is {2, 9, A, B, S}.

a) Draw a deterministic finite automata (DFA) that accepts the strings as NieR: Automata. Handle all transitions. Assume there is no more input once an accepting state is reached.

b) Using the DFA from part (a), show the states traversed when the DFA reads the string `2B92SAB9A`.

c) Pascal accidentally let malware infext NieR: Automata. For some reason, it now does Boyer-Moore string matching for pattern `92BAS2B`. Draw the suffix skip array.

Suffix Trees

a) Draw the suffix tree for the string `LOPHOPHORA`.

b) The longest repeated substring is the longest substring of a string that occurs at least twice. For example, the longest repeated substring of `LOPHOPHORA` is `OPHO`. Given the suffix tree for a string T of length n, design an algorithm to find the longest repeated substring of T in $O(n)$ time.

Burrows-Wheeler Transform

The following key was encoded by the Burrows-Wheeler Transform.

`EPESLPP$ASEAR`

Decrypt it using the method outlined in the slides, showing the array of tuples A, $sort(A)$, and each value of j.

B-Trees

Consider the following B-Tree, of order 5:

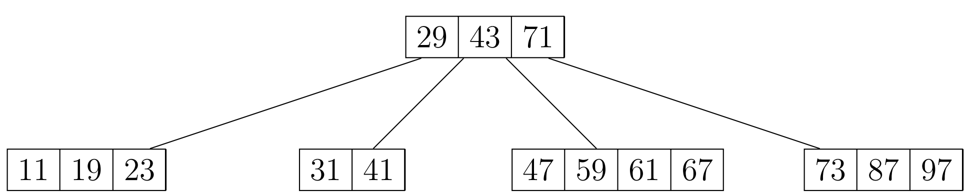

Figure 2: B-Tree of order 5

a) Insert the following keys into the B-Tree, in the order given: 13, 53, 17. Show the tree after each insertion.

b) Delete the following keys from the original B-Tree, in the order given: 19, 43, 31, 29. Show the tree after each deletion.

Range Query

Consider an array A of n integers. We want to implement a range query called $MaxDiff(i, j)$ which will find the maximal difference between two elements from $A[i]$ to $A[j]$ inclusive, for $i < j$. For example, if you run the query $MaxDiff(3, 7)$ on the array below:

$A = 5\ 0\ 2\ 8\ 9\ 4\ 6\ 7\ 6\ 1\ 3$

Between indices 3 and 7 in the array above, the largest number is 9 and the smallest number is 4, so the maximal difference is $9 - 4 = 5$.

Using a data structure with space complexity $O(n)$, implement $MaxDiff(i, j)$ to answer queries in $O(\log n)$ time. There are no limits on the runtime for preprocessing the array into the data structure, but it should not be a randomized algorithm.