University of Waterloo
CS240R Fall 2018
Help Session Problems

Reminder: Final on Friday, December 7 2018

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

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

g) Inserting a set of keys into an empty 2-3 Tree will always result in the same final tree regardless of the insertion order.

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

i) If the root of a binary tree has balance -2 while both children are AVL trees, then calling AVL-fix will decrease the height of the tree.

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

Multiple Choice

Pick the best answer for each question.

1. Which of the following functions $f(i)$ would cause interpolation search to have the least worst-case runtime on an array A with $A[i] = f(i)$?

   a) $f(i) = \log(i)$
   b) $f(i) = i$
   c) $f(i) = i^2$
   d) $f(i) = 2^i$

2. The minimum number of nodes in a 2-3 tree of height 2 is:

   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. Which of the following search operations on a non-dictionary structure has the most efficient worst-case runtime?

   a) Searching for a specific key in a max-heap.
   b) Searching for a specific point in a kd-tree.
   c) Searching for any occurrence of a specific character in a text using a suffix tree.
   d) Searching for a specific character in a decoding trie of characters.

8. CS240 is a course about

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

Hashing

Let $p \geq 3$ be prime, and consider the universe of keys $U = \{0, 1, \ldots, p^2 - 1\}$. Answer each question for an initially empty hash table of size p.

a) Using double hashing with $h_1(k) = k \bmod p$ and $h_2(k) = \lfloor k/p \rfloor + 1$, give a sequence of **two** keys to be inserted that results in failure.

b) Using cuckoo hashing with $h_1(k) = k \bmod p$ and $h_2(k) = k \bmod (p-1)+1$, give a sequence of **three** keys to be inserted that results in failure.

c) Using cuckoo hashing with $h_1(k) = k \bmod p$ and $h_2(k) = \lfloor k/p \rfloor + 1$, give a sequence of **three** keys to be inserted that results in failure.

Pattern Matching

Consider the problem of searching for the pattern $P = \texttt{OMNOMNOM}$ in the text $T = \texttt{OMNOONOMNEMOMNOMNOM}$ with the alphabet $\Sigma = \{O, M, E, N\}$.

a) Construct the failure array for P. Then use KMP to search for P in T, showing each character comparison. Indicate characters that are known to match due to the failure array using parentheses.

b) Construct the last-occurrence function and suffix-skip array for P. Then use Boyer-Moore to search for P in T, showing each character comparison. Indicate characters that are known to match (due to either the suffix-skip array or the last-occurrence function) using parentheses.

You may use the following table as a template. Use a separate table for each sub-problem, and add rows whenever necessary.

O	M	N	O	O	N	O	M	N	E	M	O	M	N	O	M	N	O	M

Skip Lists

a) Given a skip-list L, design an algorithm to return the last key-value-pair of the skip-list, i.e. the node at the bottom level with the largest key, excluding the special $\pm\infty$ keys. The algorithm should run in $O(\log n)$ expected time.

b) Modify the skip-list structure so that the last key-value-pair can be found in $O(1)$ time. Modifications to the search, insert, or delete operations should ensure that each of the operations still run in $O(\log n)$ expected time.

Huffman Compression

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

0010000111010101110001011010010

Char	(space)	: (colon)	d	ℓ	p	s	u	w
Code	100	1011	1010	010	001	000	11	011

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

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

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 in Figure 1.

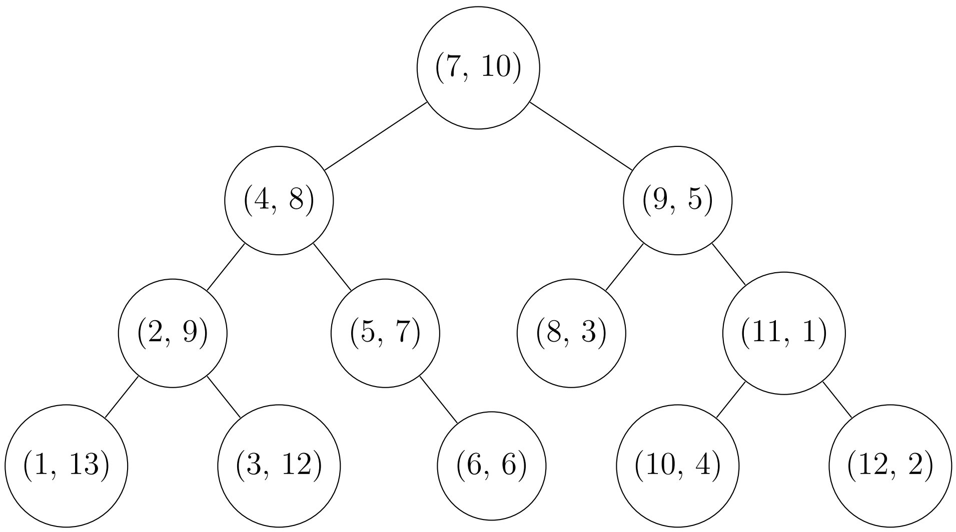

Figure 1: Range Tree x-BST

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.

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. Assume that branches are ordered as \$, 0, 1. 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.

Suffix Trees

Sajed discovered a secret message in the form of a Suffix Tree S, indicating the location of a hidden treasure.

a) Design an algorithm that recovers the original text T from its corresponding suffix tree S. The algorithm should run in $O(n)$ time while using $O(n)$ auxiliary space.

b) Determine the original text for the following suffix tree:

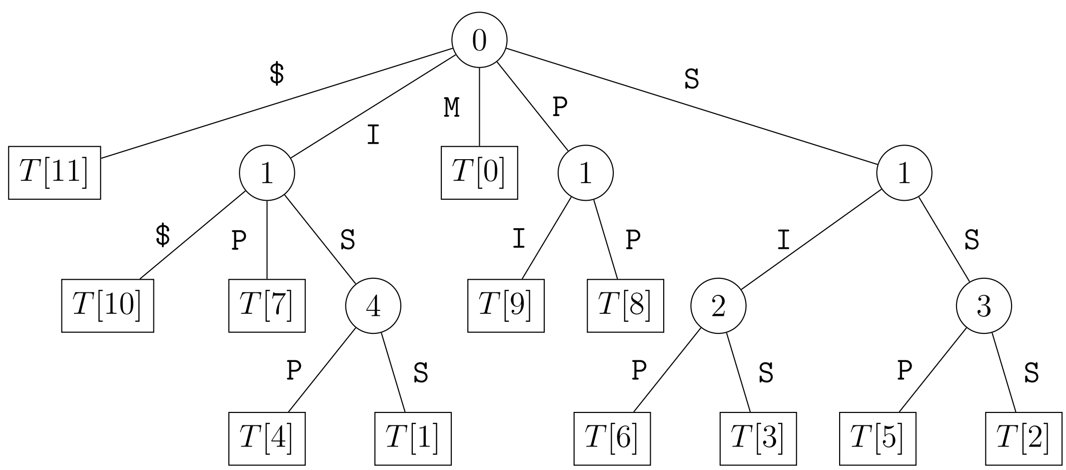

Figure 2: Mysterious Suffix Tree

Lempel-Ziv-Welch Encoding

Encode the following string using LZW compression:

`DARK_DAN_BARKS_DANK`

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

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

String Decoding

The following bit-string C was generated by 3 steps of encoding: BWT, MTF, RLE.

$$C = 0010010001101111110001000011111100011111010010010110101100111$$

a) The final step of encoding was applying RLE to encode C' to C. Use run-length decoding to recover C'.

b) The middle step of encoding was applying MTF to encode S' to C', using the following initial dictionary:

Character	\$	␣	...	A	L	M	N	O	P	...	Z
Code (DEC)	0	1	...	6	17	18	19	20	21	...	31

The codewords here are shown in decimal, but are each represented using 5 bits of binary. For example, the decimal codeword 7 would be represented as 00111. This ordering of the characters is also the sorted ordering.

Decode the C' from part (a) using MTF to show that $S' =$ `AAPP␣OOOL$MM`.

c) The first step of encoding was applying BWT to encode S to S'. Apply the inverse BWT on S' to recover the original text S.

B-Trees

Consider the following B-Tree, of order 5:

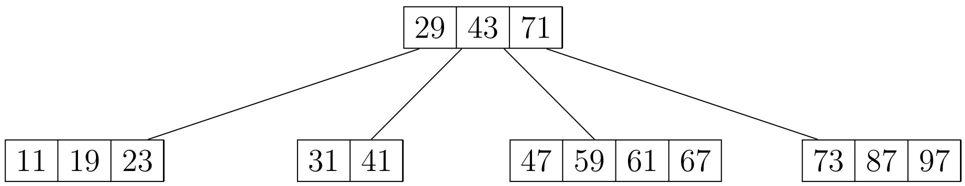

Figure 3: B-Tree of order 5

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, suppose our array A is:

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

If we run the query $MaxDiff(2, 9)$, then the subarray from indices 2 to 9:

$A[2 \ldots 9] = 5\ 4\ 5\ 6\ 3\ 4\ 5\ 7$

The largest number is 7 and the smallest number is 3, so the maximal difference is $7 - 3 = 4$. The query $MaxDiff(2, 9)$ should return 4.

Design a data structure for A with space complexity $O(n)$ to answer queries of the form $MaxDiff(i, j)$ 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.