

Final Practice Problems

Reminder: Final Exam is at either December 21, 4 pm, or December 22, midnight, depending on your chosen timeslot.

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 exam, and should not be used as a reference for its content.

1 True/False

For each statement below, write true or false.

- a) Open addressing hashing that uses linear probing will require two hash functions.
- b) Suffix trees for pattern matching require preprocessing the pattern.
- c) Binary Search and Interpolation Search are comparison-based searching algorithms.
- d) When doing range search on a quadtree, if there is no point within the range specified, the worst case runtime complexity is in $\Theta(h)$.
- e) The best-case expected runtime for Karp-Rabin pattern matching is in $O(m)$.
- f) Every string consisting of English characters and one end-character can be decoded with inverse BWT.
- g) When using KMP to search for the pattern $\mathbf{a}^{m-1}\mathbf{b}$ in the text \mathbf{a}^n , the positions of the pattern shifts are the same as the brute-force algorithm.
- h) If $\alpha = 1$ for hashing with chaining, inserting a new key will always fail.
- i) Every AVL Tree is also a 2-4 Tree.
- j) Move-to-front transformation utilizes an adaptive dictionary.

2 Multiple Choice

Pick the best answer for each question.

- a) Which of the following functions $f(i)$ would cause interpolation search to have the least worst-case runtime on an array A with $A[i] = f(i)$?
 - i) $f(i) = \log(i)$
 - ii) $f(i) = i$
 - iii) $f(i) = i^2$
 - iv) $f(i) = 2^i$

- b) Given $h_0(k) = k \bmod 7$ in a hash table of size 7, which of the following hash functions would be most suitable for h_1 in double hashing?
- i) $h_1(k) = k^2 \bmod 7$
 - ii) $h_1(k) = (k \bmod 6) + 1$
 - iii) $h_1(k) = 2 \cdot (k \bmod 4)$
 - iv) $h_1(k) = \lfloor \frac{1}{2} \cdot (k \bmod 13) \rfloor$
- c) Given $h_0(k) = k \bmod 7$ with two hash tables, each of size 7, which of the following hash functions would be most suitable for h_1 in cuckoo hashing?
- i) $h_1(k) = k^2 \bmod 7$
 - ii) $h_1(k) = (k \bmod 6) + 1$
 - iii) $h_1(k) = 2 \cdot (k \bmod 4)$
 - iv) $h_1(k) = \lfloor \frac{1}{2} \cdot (k \bmod 13) \rfloor$
- d) If the root of a quadtree represents the region $[0, 128) \times [0, 128)$ while the deepest (lowest) internal node represents the region $[88, 92) \times [24, 28)$, what is the height of the quadtree?
- i) 4
 - ii) 5
 - iii) 6
 - iv) 7
- e) Which one of the following statements about compressed tries is false?
- i) Every internal node stores an index indicating the bit position to be tested on a search.
 - ii) The root of the compressed trie always tests the first bit.
 - iii) A compressed trie that stores n keys always contains less than n internal nodes.
 - iv) The height of a compressed trie never exceeds the length of the longest string it stores.
- f) Given that P is not in T , and $m = \sqrt{n}$, which of the following pattern matching algorithms would have the lowest best-case runtime for searching (excluding preprocessing) for P in T ?
- i) Rabin-Karp
 - ii) Knuth-Morris-Pratt
 - iii) Boyer-Moore Bad Character
 - iv) Suffix Array
- g) CS240 is a course about
- i) Data structures and algorithms
 - ii) Unreasonable time management
 - iii) Reconsidering academic/life choices
 - iv) All of the above

3 Hashing

Let $p \geq 3$ be prime, and consider the universe of keys $U = \{0, 1, \dots, p^2 - 1\}$.

- With a hash table of size p , and using double hashing with $h_0(k) = k \bmod p$ and $h_1(k) = \lfloor k/p \rfloor + 1$, give a sequence of **two** keys to be inserted that results in failure.
- With two hash tables of sizes p and $(p - 1)$, and using cuckoo hashing with $h_0(k) = k \bmod p$ and $h_1(k) = k \bmod (p - 1)$, give a sequence of **four** keys to be inserted that results in failure.
- With two hash tables of size p each, and using cuckoo hashing with $h_0(k) = k \bmod p$ and $h_1(k) = \lfloor k/p \rfloor$, give a sequence of **six** keys to be inserted that results in failure.

4 Huffman Compression

- 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.

- Agent Davies doesn't know the information in the message beforehand, but upon seeing the decoded string, she immediately realizes that the message has been tampered with. Explain how Sylvie determined this.

5 Boyer-Moore

Boyer-Moore can be modified in many ways. For each of the modifications listed below, state whether or not the modification is valid, i.e., the modified Boyer-Moore will always successfully find the first occurrence of P in T , if P appears in T , or return FAIL if P is not in T . If the answer is "Yes", provide a brief explanation of why it is still valid. If the answer is "No", demonstrate a counterexample, i.e., trace the algorithm on a specific P and T of your choice where the result is incorrect.

Note: We are using Boyer-Moore with only the Bad Character heuristic here.

- Using a first-occurrence function (denoting the index of the first occurrence of the argument character) instead of a last-occurrence function.
- When checking a pattern shift, compare characters from the start of the pattern and move forward, instead of scanning backwards from the end of the pattern.
- Use the last-occurrence function for $P[0 \dots m - 2]$, i.e., P with its last character removed, instead of the last-occurrence function for P .

6 Compressed Trie Height

Let T be a compressed trie of n strings from an alphabet Σ , where the end-character is not in Σ . Give exact expressions (not order notation) for the following, in terms of n and $|\Sigma|$:

- What is the maximum height of T ?
- What is the minimum height of T ?

7 Karp-Rabin

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

$$h(P) = (\# \text{ of occurrences of } \mathbf{A}) + 2 \times (\# \text{ of occurrences of } \mathbf{C}) \\ + 3 \times (\# \text{ of occurrences of } \mathbf{G}) + 4 \times (\# \text{ of occurrences of } \mathbf{T})$$

- Given the pattern $P = \mathbf{TAGCAT}$ and sequence $T = \mathbf{TGCCGATGTAGCTAGCAT}$, use the table below to show all the character comparisons performed during Karp-Rabin 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 Karp-Rabin problem.

- 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.
- Give a length- m pattern P and a length- n sequence T , with $n > m$, such that the Karp-Rabin algorithm achieves $\Omega(nm)$ character comparisons.

8 KD-Trees

Consider the following set of 3-dimensional 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) Mark the nodes which are visited during a range query in the rectangular box $[60, 70] \times [90, 100] \times [50, 90]$.

9 Suffix Arrays

Agent Li is trying to acquire some secret text T , consisting of uppercase English letters only, but all he was able to obtain was a randomly scrambled text T' (which is a permutation of T), and the suffix array A^S of the original T . Design an algorithm to utilize T' and A^S to recover the original text T in $O(n)$ time.

10 Run-Length Encoding

For the following questions, you may assume that n is divisible by 4.

- a) For each $n > 0$, give a string of n bits that achieves the worst compression ratio with Run-Length Encoding from all n -bit strings, and state the exact compression ratio achieved.
- b) Same question, but for the best compression ratio. **Hint:** If there are multiple runs in the string, would it help to modify the string by merging two or more runs into a single run?

11 Consecutive Trie Strings

Given an uncompressed trie T that stores a list of binary strings, design 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 pre-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.

12 Lempel-Ziv-Welch

- a) Suppose a text of length n is encoded with LZW into a sequence of w codewords. Assume that the number of bits to represent each codeword is large enough such that the dictionary never becomes full (i.e., a new substring is always added to the dictionary after every step). Give a $\Theta()$ bound in terms of n to denote the smallest value of w .
- b) A secret message was reliably intercepted as follows (spaced for convenience):

00011001 00100001 01000000 00101000 00100001 00011000 00100001 01000000 00001011 00001011

Intelligence from other sources indicated that this message was generated by LZW encoding with 8-bit codewords, where the initial dictionary has size 64, though the initial dictionary itself is unknown.

Agent Zhang does not know the original text or the initial dictionary, but he immediately realized that their intelligence was false. Explain how Jason was able to determine this.

13 Burrows-Wheeler Transform

For the following questions, assume $k > 0$ and $\ell > 0$.

- a) Consider the string S of the form $(AH)^k A$ (e.g., AHAHA for $k = 2$). What is the Burrows-Wheeler Transform for this string?
- b) Consider the string S of the form $H^k A^\ell$ (e.g., HHHAAAA for $k = 3, \ell = 4$). What is the Burrows-Wheeler Transform for this string?
- c) Consider the string S of the form $A^k H^\ell$ (e.g., AAAAAHH for $k = 5, \ell = 2$). What is the Burrows-Wheeler Transform for this string?

14 B-Trees

Consider the following B-Tree, of order 5:

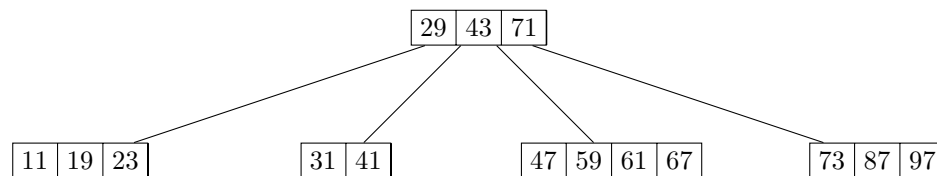


Figure 1: B-Tree of order 5

- 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. When deciding between successor/predecessor, choose the successor. When deciding between left or right sibling for transfer/merge, select the right sibling. Show the tree after each deletion.

15 Maximal Difference

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 is:

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

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 constraints on the runtime for preprocessing the array into the data structure.