University of Waterloo
CS240 Winter 2025
Assignment 5

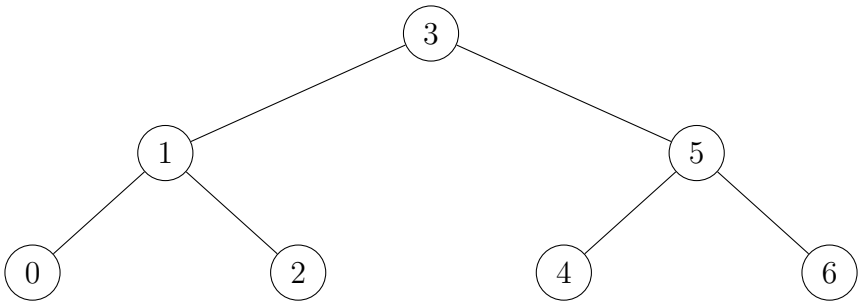
Due Date: Tuesday, April 1 at 5:00pm

Please read `https://student.cs.uwaterloo.ca/~cs240/w25/assignments.phtml#guidelines` for guidelines on submission. **Each question must be submitted individually to Crowdmark.** Submit early and often.

Grace period: submissions made before 11:59PM on April 1 will be accepted without penalty. Your last submission will be graded. Please note that submissions made after 11:59PM **will not be graded** and may only be reviewed for feedback.

Question 1 [3+4+3=10 marks]

a) Consider the BST below.



List all nodes that could be classified as the topmost inside node during some range search. For each node that can be a topmost inside node, give the range search which results in the node labeled as the topmost inside node. For every node that cannot be a topmost inside, explain why.

b) Draw a range tree that corresponds to the following set of 2D points:

$$S = \{(3,4),(10,2),(9,9),(5,7),(6,1),(0,3),(1,8)\}.$$

Draw the primary tree, and then all the associate trees. Make the primary tree and all the associate trees of the smallest height possible. Make sure to indicate to which node v an associate tree $T(v)$ belongs to. No explanation is needed.

c) Consider a relaxed version of the range tree where neither the primary tree, nor the associate trees are required to be balanced. What is the worst case running time of range search in such a relaxed range tree, assuming that the output size s is $O(1)$? Explain. State your answer in terms of Θ notation.

Question 2 [4 marks]

Let $M = 7$ be the prime number chosen by Rabin-Karp, $P = 118$ be the pattern to search for, and $h(k) = k \bmod M$. Give a text T, with length 9 that produces the worst-case number of comparisons for Rabin-Karp fingerprinting. T should not contain pattern P. Explain why T produces the worst-case.

Question 3 [4+4+2=10 marks]

a) Compute the KMP failure array for the pattern $P = bbbabb$. If you wish, you can show the intermediate steps, but providing the final answer is sufficient.

b) Show how to search for pattern $P = bbbabb$ in the text $T = bbbcbbacbbbbbabb$ using the KMP algorithm. Indicate in a table such as Table 1 which characters of P were compared with which characters of T, like the example in Module 9. Place each character of P in the column of the compared-to character of T. Put round brackets around characters if an actual comparison was not performed. You may need to add extra rows to the table.

b	b	b	c	b	b	a	c	b	b	b	b	b	a	b	b

Table 1: Table for KMP problem.

c) Assume the alphabet consists of characters $\{a, b, c\}$. Construct a pattern of length 10, such that if we run KMP algorithm and there is a mismatch at any $0 < j < 10$, the pattern moves forward as much as possible (i.e. the new guess is as large as possible). Your pattern must start with 'a'. Explain your answer.

Question 4 [3+6=9 marks]

In this question, we will develop a version of Boyer-Moore algorithm which expands the bad character heuristic by using 'next-to-last' occurrence array, in addition to the last occurrence array.

a) Define $N(c)$ as the next-to-last occurrence of character in pattern P. For example, if $P = aabra$, then $N('a') = 1$. Design an algorithm to compute N in $O(m + \Sigma)$ running time, where m is the length of pattern P. If there is no second to last occurrence of a character, define $N(c) = -1$. You can describe your algorithm either in pseudo-code or in English. Briefly justify the running time and correctness of your algorithm (one sentence for correctness and one sentence for the running time).

b) Develop a modified version of Boyer-Moore algorithm that makes use of both the last occurrence array L and the next-to-last occurrence array N. If text and pattern characters match, just as in the standard Boyer-Moore, your algorithm must decrease both i and j. Whenever a mismatch occurs, your algorithm should make the largest possible valid[1] shift of the pattern, based on the information in L and N. You must provide pseudo-code for your algorithm as well as a justification for your modifications. Your pseudocode should be named as $BoyerMooreModified(T, P, L, N)$, where the input parameters, respectively, are the text, pattern, the last and next-to-last occurrence arrays.

Question 5 [2+3+5=10 marks]

a) Construct the suffix array for $S = morduspordus$. You can show the intermediate steps, but showing the final answer is sufficient.

b) Let $T = (x_1x_2)^n$, where the character x_1 is less than the character x_2 and raising to power n means repeating n times. For example, if $x_1 = a, x_2 = b$ and $n = 3$, then $T = ababab$. Let A be the suffix array of T. Which index of A stores the suffix of T such that this suffix starts with the earliest occurrence of pattern x_1x_2 in the text?

c) You are given text T and its suffix array A. Explain how to construct an AVL tree such that this AVL tree, in conjunction with the suffix array A, can be used to find the leftmost occurrence of pattern P in T in $O(m \log n)$ time, where n is the length of the text and m is the length of the pattern.

Question 6 [3+2=5 marks]

a) Build the Huffman's tree for the string `AAABBCCCCCD` and give the coded text. When building the tree, put the smaller weight subtree to the left. If subtrees have the same weight, put the tree which has the character earlier in the alphabet to the left. Show all the intermediate steps.

b) Prove that in the Huffman's tree, every leaf at the deepest level must have a sibling (i.e. the parent of any leaf at the deepest level has another child).

[1]Here valid means that while you make the largest shift, you do not skip any shifts where a match is possible.