

# University of Waterloo

## CS240 Spring 2024

### Assignment 5

**Due Date: Tuesday, July 23 at 5:00pm**

Please read the following link for guidelines on submission:

<https://student.cs.uwaterloo.ca/~cs240/w24/assignments.phtml#guidelines>

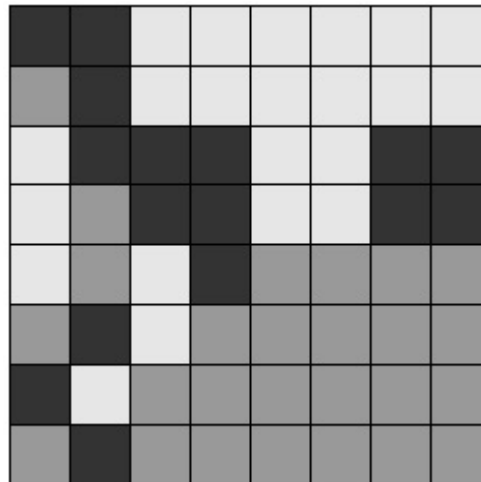
**Each question must be submitted individually to MarkUs as a PDF** with the corresponding file names: a5q1.pdf, a5q2.pdf, .... It is a good idea to submit questions as you go so you aren't trying to create several PDF files at the last minute.

**Late Policy:** Assignments are due at **5:00pm**, with the grace period until 11:59pm.

#### Problem 1 Quadtrees [3+4=7 marks]

For all parts of this question, use the convention that each internal node of a quadtree has exactly four children, corresponding to regions NE, NW, SW and SE, in that order.

- a) One application of quadtrees is image compression. An image (picture) is recursively divided into quadrants until the entire quadrant is only one colour. Using this rule, draw the quadtree of the following image. There are only three colours (shades of grey). For the leaves of the quad tree, use 1 to denote the lightest shade, 2 for the middle shade and 3 for the darkest shade of grey.



- b) Another application is to compare two images. Given two black and white images (i.e. each pixel of the image is either 0 or 1) each of size  $2^k \times 2^k$  stored as quadtrees, give an algorithm that returns the quadtree for the Intersection operation: if corresponding pixels in both images is 1, then their intersection is 1; otherwise 0.

Runtime analysis is not required but your algorithm should be as efficient as possible; i.e. marks may be deducted for terribly inefficient implementations.

**Problem 2 [3+5=8 marks]**

- a) Draw the kd-tree representing the set of 2D points

$$S = \{p_1, \dots, p_8\} = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8)\}.$$

- b) Give an algorithm using pseudocode for finding a point with the smallest x-coordinate in a kd-tree storing 2D points. Your algorithm should be as efficient as possible. State and explain the worst case running time of your algorithm.

**Problem 3 Range Trees [2+1+2+2=7 marks]**

- a) Draw a 2-dimensional range tree of minimal height for the following set of points:

$$\{(7, 88), (12, 19), (22, 33), (27, 29), (28, 9), (31, 99), (42, 66)\}$$

- b) Suppose a two dimensional range tree data structure stores  $n$  points, and that the BST ordered by  $x$ -coordinates is perfect, i.e., every level is completely filled. Give an exact closed form formula in terms of  $n$  for the sum of the number of nodes in the  $x$ -ordered BST plus the total number of nodes in all  $y$ -ordered BSTs.
- c) Assume that we have a set of  $n$  numbers (not necessarily integers) and we are interested only in counting the number of points that lie in a range rather than in reporting all of them. Describe how to modify a 1-dimensional range tree (i.e., a balanced BST) so that such a range counting query can be performed in  $O(\log n)$  time (independent of  $s$ ). Briefly justify that your algorithm is within the expected runtime.
- d) Next, consider the 2-dimensional case where we have a set of  $n$  2-dimensional points. Given a query rectangle  $R$ , suppose we only want to find the number of points inside  $R$ , rather than the points themselves. Explain how to modify the Range Tree data structure and the search algorithm such that counting queries can be performed in  $O((\log n)^2)$  time. Briefly justify that your algorithm meets the runtime requirement.

**Problem 4 KMP [2+3=5 marks]**

- a) Compute the failure array for the pattern  $P = \text{bababc}$ .
- b) Show how to search for pattern  $P = \text{bababc}$  in the text  $T = \text{bacbbabbabababcbacbb}$  using the KMP algorithm. Indicate in a table such as Table 1 which characters of  $P$  were compared with the characters of  $T$ . Follow the example on slide 16 in module 9. Place each character of  $P$  in the column of the compared-to character of  $T$ . Place parentheses around the character if an actual comparison was not performed. You may not need all space in the table.

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

Table 1: Table for KMP problem.

**Problem 5 Boyer-Moore [2+3+3=8 marks]**

- a) Construct the last occurrence function  $L$  for pattern  $P = badodadb$  where  $\Sigma = a, b, c, d, o, t$ .
- b) Trace the search for  $P$  in  $T = adtbadtbadtadtbadodadbadt$  using the Boyer-Moore algorithm. Indicate in a table such as Table 2 which characters of  $P$  were compared with which characters of  $T$ . Follow the example on slide 25. Place each character of  $P$  in the column of the compared-to character of  $T$ . Put brackets around the character if they are known to match from the previous step (similar to the example in the slides). Use a new row when sliding the pattern. You may not need all rows in the table. Add more rows to the table if you need more.
- c) For any  $m \geq 1$  and any  $n \geq m$ , give a pattern  $P$  and a text  $T$  such that the Boyer-Moore algorithm “looks at” (compares a character of the text with a character of the pattern) exactly  $\Theta(n/m)$  characters. Justify your answer.

**Problem 6 Suffix Trees and Arrays [3+2=5 marks]**

- a) Draw the suffix tree corresponding to the text  $T = abracadabra$ .
- b) Construct suffix array for  $S = balbes$ .

a	d	t	b	a	d	t	b	a	d	t	a	d	t	b	a	d	o	d	a	d	b	a	d	t

Table 2: Table for Boyer-Moore problem.