University of Waterloo
CS240 Fall 2021 Final Exam

Time Slot 1: Tuesday, December 21 at 4pm to 8pm
Time Slot 2: Wednesday, December 22 at 12:01am (Midnight) to 4:01am

Based on the time slot you selected, your exam will start at either 4pm or 12:01am (Midnight), Waterloo, Ontario time. At that time, you may download the exam. All exams start at exactly 4pm or 12:01am (Midnight), depending on your time slot, whether you download the exam then or not. **No extra time will be given for students starting late.** End time is as stated for each sitting. Students registered with AccessAbility Services for extra exam time should check the following link to see their end time:
https://student.cs.uwaterloo.ca/ cs240/cgi-bin/displayFinalTime.cgi
You have been allocated 2.5 hours to write the exam and an additional 1.5 hours to create a file or individual files (**one file per question**) and to submit them to Crowdmark.

Exam Procedure: After 2.5 hours, you must create and upload the work that you have completed to Crowdmark. If you find that you still have time remaining, you may then go back and work on the exam and submit an updated file but make sure to do so before time has expired.

Late Submissions: You are given 1.5 hours to create and upload the necessary files to Crowdmark - this is ample time to do so. Work submitted after the 4 hour deadline is viewed as an attempt to gain an advantage over other students and will incur a late penalty.

To accommodate any small differences in time, we give a grace period (no penalty) of 2 minutes. If you submit anything more than 2 minutes late but less than 15 minutes late, you will incur a 10% exam deduction; i.e. 10% will be deducted from your final exam grade. After 15 minutes, no submissions will be accepted.

In most cases, this means that if you are submitting a single question late, you will lose more marks for the late submission than the question you are submitting is worth. By submitting after the deadline you are accepting that you understand the late policy and accept this penalty.

Also, failure to upload questions to Crowdmark (by sending them by email to course personnel) is considered trying to gain a time advantage and will incur a 10% penalty deducted from your final exam grade. Note that if they are emailed to course personnel and late, 20% is deducted from your final exam grade.

Piazza: Public posts to Piazza will be turned off during the exam, so only private posts to instructors will be allowed. In the case where you think you have found an error or lack of clarification, read the question again carefully first, then make a post if you still have a question. We will put any errors or clarifications on Piazza in an Official Final Exam Clarifications post.

Academic Integrity Declaration: Your Academic Integrity Declaration is to be submitted to **MarkUs** where we have means to autocheck correct completion.

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. You must read and sign an Academic Integrity Declaration FINAL-AID.txt (found on the course Testing/Exams webpage) before starting work on the assessment and submit it to **MarkUS before the deadline of December 22nd at 4:01am** (i.e. **read, sign and submit now or as soon as possible) or your assessment will not be marked**. The agreement will indicate what you must do to ensure the integrity of your grade.

Submitting to Crowdmark: Your answers to the assessment questions are to be submitted to Crowdmark. After the assessment has started, you will receive email from the Crowdmark mailer with the link for submitting your files (if you do not see it, check your spam and junk folders in your email reader).

Submitting to Crowdmark may be done as separate question files (one for each question) **or** *one* PDF, **as long as**, each question answer starts on a new page and you can drag page by page to the correct question in Crowdmark. Crowdmark accepts PDF, JPG, and PNG files. The size limit is 12MB per JPG or PNG file and 25MB per PDF file.

Make sure each page has the question number and part labels (if any). You may use LaTeX to create your file or hand write your answer and scan it or take a picture.

Generally speaking, if your answer is longer than a page for a question, it is too long. However, you can still submit more than one page for each question, if necessary (for example, if your writing is large or you include a drawing).
More complete instructions for submitting and verifying submission can be found at:
https://crowdmark.com/help/completing-and-submitting-an-assignment/
https://crowdmark.com/help/verifying-that-an-assignment-was-submitted/

Problem 1 [1+2+2=5]

a) Give the run length encoding for 01001111.

b) The Oracle will answer any Yes/No question you ask; such as "Will I pass CS 240?". They respond with the following encoded answer: 01101011.
Luckily, you have also been provided the following codes:

$$E = 1010, S = 11, O = 1011, Y = 01, N = 0110.$$

Briefly explain the problem with this encoding and how it can be improved.

c) Consider a quadtree and kd-tree containing the same n 2-dimensional points for some large value of n. Is one of the trees guaranteed to be shorter than the other? Briefly justify your answer.

Problem 2 [2+3=5]

a) Consider using MSD radix sort to sort n integers whose values range from 1 to n^n (inclusive). Assuming you choose the radix, is it possible to have a worst-case runtime of $o(n^2)$? Briefly justify your answer.

b) Suppose you are given a text T of length n and a queue of k patterns P where each pattern in P has length m. The Karp-Rabin algorithm is then used to search for each pattern in the queue using only a single pass through the text. You may assume that a random prime has been chosen and that the hash value of each pattern has been calculated and stored with each pattern in the queue. At each step, when a hash value of the text is computed, each pattern in the queue is compared with the text. If a match is found, the text index and pattern are printed and the pattern is removed from the queue; otherwise, the pattern is moved to the back of the queue. Karp-Rabin moves to the next block of text when all patterns in the queue have been checked. Suppose that $\sqrt{k}$ patterns are found. Give the best-case expected runtime and worst-case runtime.

Problem 3 [3+3=6] Analysis I

a) Prove from first principles that $7n + 3$ is $o(n \log n)$.

b) Give a Θ expression for the runtime of the following code. Briefly justify your answer.

```
Mystery(n)
    c ← 0
    i ← 0
    s ← 1
    while i ≤ n do
        c ← c * 1
        i ← i + s
        s ← s + 2
        for j ← (n − 10000) to n do
            j ← j + s
```

Problem 4 [3+3=6] Analysis II

a) Consider the following algorithm.

```
RandomRoller(n)
1.  i := random outcome from rolling the die (see below)
2.  if i is even do
3.      for j ← 1 to n do
4.          write * to output
5.  else
6.      for j ← 1 to i do
7.          write * to output
8.      RandomRoller(n)
```

The die used in the function above on line 1 is a 5-sided die (with numbers 1 2 3 4 5) where each outcome is equally likely. Give a recurrence relation $T(n)$ for the expected number of asterisks (*) that will be printed and simplify $T(n)$ to be only in terms of n.

b) Let $D = [a_1, a_2, \ldots, a_n]$ be a dictionary implemented with the Move-To-Front (MTF) heuristic that stores n distinct, unordered items in a linked list L. A sequence of m searches is performed on keys present in D. The number of distinct keys among the m searches is k.

Determine an exact expression for the worst-case number of key comparisons to perform all m searches (in terms of m, n, k where appropriate) and then give a simplified

Θ notation expression. Describe the search sequence that produces your expression to justify your answer.

Problem 5 [3+3=6]

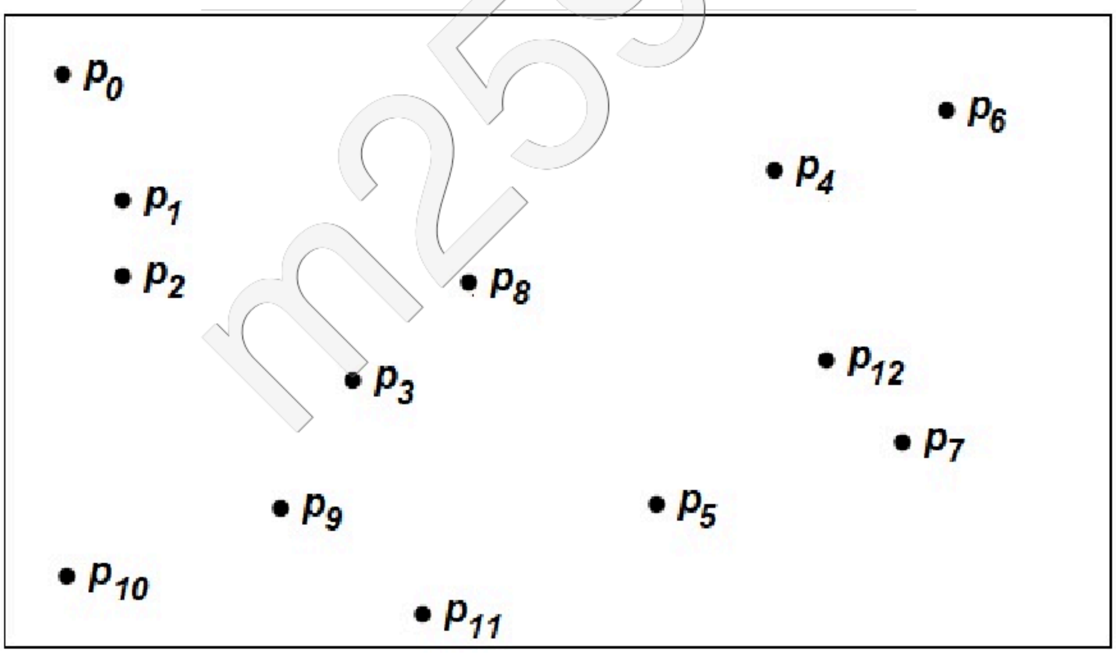
Suppose you are given an array A of size n where n is even. A is ordered so that the first half contains integers that are in increasing order and the second half contains integers that are in decreasing order. For some instances of A, the integers in one or both halves of A are uniformly distributed and in other instances they are not. However, the distribution of inputs is not known.

a) Give an algorithm to search A for a given integer x (which may or may not be in the array). Briefly explain the runtime complexity of your algorithm.

b) Suppose the integers in each half are uniformly distributed. Describe an algorithm to find the 13th smallest integer. Briefly explain the runtime complexity of your algorithm.

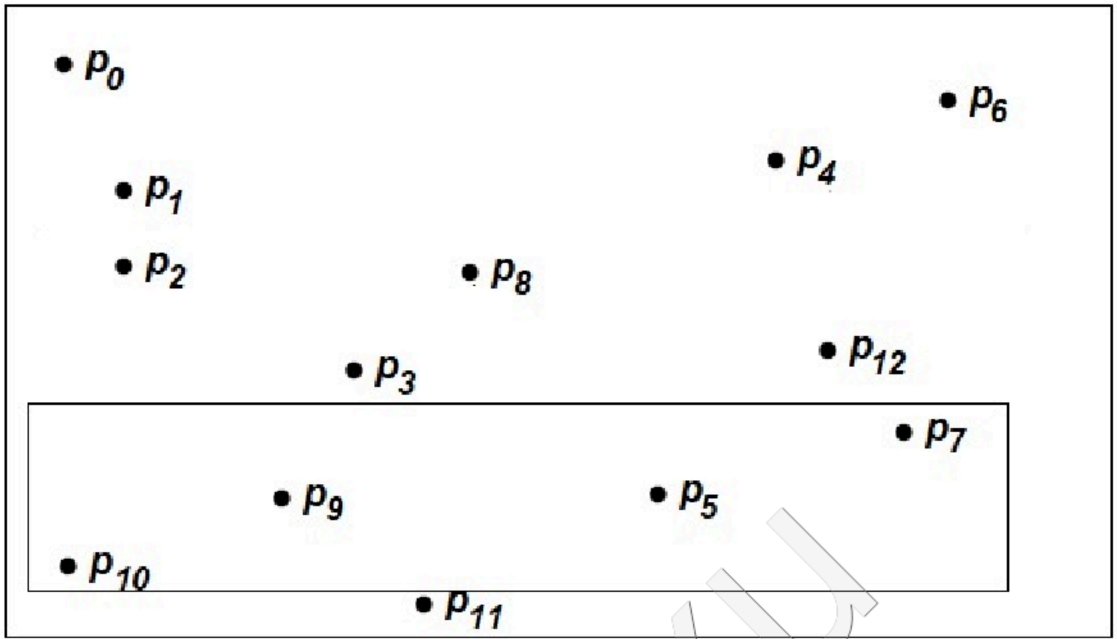
Problem 6 [3+3=6] kd-Trees

Given the following set of 2-dimensional points:



a) Draw the kd-tree (not the plane-partition) for the points above.

b) Consider the query rectangle shown below. Draw the subset of the tree from part a) that will be examined in a range search for this region.


p_0
p_6
p_4
p_1
p_2
p_8
p_{12}
p_3
p_7
p_9
p_5
p_{10}
p_{11}

Problem 7 [2+4=6] Pattern Matching

a) Boyer-Moore (BM) and Knuth-Morris-Pratt (KMP) both perform a character by character search for a pattern in text. What is the main difference in the order that they search the text and why is this difference an advantage for Boyer-Moore?

b) Consider the pattern P=**AKAKAMA** over the alphabet $\Sigma = \{A, K, M, R\}$.

- **i)** Give the KMP failure array for P.
- **ii)** Give the BM last occurrence array for P.
- **iii)** Give the Suffix Array for P.

Problem 8 [3+1] Huffman Coding

a) Construct the Huffman code for the word **MOMLATEXTATTOO**.
Draw the tree and separately list the code given for each letter; i.e. for example:
A: code for A
E: code for E . . .

Convention for building the structure: Place lower-valued structure on the left. In case of a tie, choose the smallest-alphabetical letter or structure containing the smallest-alphabetical letter. Also, (if tied) place the smallest-alphabetical letter or structure containing the smallest-alphabetical letter on the left.

b) Using the Huffman tree constructed in part a), decode the following binary string:

`11010110010001110001000001000010`

Problem 9 [2+2+2=6] Burrows-Wheeler Transform

a) Compute the Burrows-Wheeler transform of `VAMPMAMA$` where `$` is the end-of-text marker.

b) Decode the follow string: `U$ULTUCHHH`

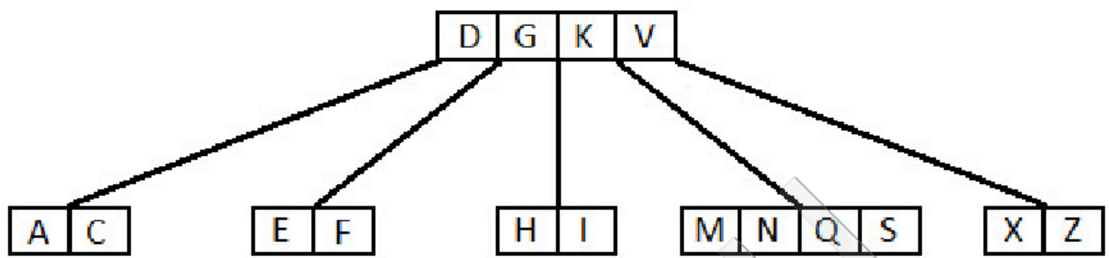
c) Briefly explain why the Burrows-Wheeler transform is useful for compression.

Problem 10 [2+2+2=6] B-Trees

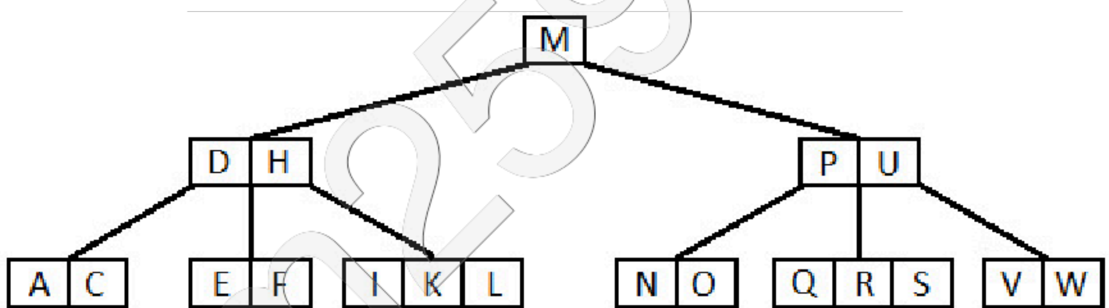
Note: The keys are letters in the English alphabet ordered in normal alphabetical order:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a) Draw the resulting tree after inserting **R** into the following (3,5)-tree.



For parts b) and c), use the following (3,5)-tree as a starting point to perform the operation specified in each part using the algorithm given in class.



b) Draw the resulting tree after deleting **C**.

c) Draw the resulting tree after deleting **N** from the B-tree given **above** part b).
DO NOT use the resulting tree from part b); i.e. your tree should contain the letter **C**.

Problem 11 [4+1+2=7] Lempel-Ziv-Welch Compression

a) Assume $k = 6$ (6-bit codes) and use the dictionary given on the last page of the exam as the initial dictionary (rather than the ASCII table).

Give the LZW compression of: **TEXTESTERTEST** followed by **EOF**.
Show the entries added to the dictionary at each step.
For convenience, output the encoding using decimal codes but state the final length of the encoding in number of bits (as if binary codes were used).

b) Provide and briefly explain a suitable compression ratio for the string above.

c) Suppose during the decoding process, that code 240 was decoded as `ALEX`, the code `EMMA` was added to the decoding table at position 342, and the next code to decode is 343.
Give the string that 343 decodes as and briefly justify how it was obtained or if it is not possible to decode 343, list some criteria about the form of strings for which it is impossible for 343 to decode as; for example, cannot decode as `A`, `AL`, `ALE` or strings with prefix `ALEX`.

See next page for the initial fixed dictionary.

Below is the initial fixed dictionary used in Problem 11 LZW.

Character	Decimal Code for LZ	6-Bit Binary Code for LZ	ASCII Code
A	0	000000	65
B	1	000001	66
C	2	000010	67
D	3	000011	68
E	4	000100	69
F	5	000101	70
G	6	000110	71
H	7	000111	72
I	8	001000	73
J	9	001001	74
K	10	001010	75
L	11	001011	76
M	12	001100	77
N	13	001101	78
O	14	001110	79
P	15	001111	80
Q	16	010000	81
R	17	010001	82
S	18	010010	83
T	19	010011	84
U	20	010100	85
V	21	010101	86
W	22	010110	87
X	23	010111	88
Y	24	011000	89
Z	25	011001	90
␣	26	011010	32
EOF	27	011011	(N/A)