**St. Francis Xavier University**
**Department of Computer Science**

# CSCI 435: Algorithms and Complexity
## Final Examination
## April 19, 2023
## 9:00am–11:30am

**Student Name:** ____________________

**Email Address:** ____________________

**Instructor:** T. J. Smith (Section 20)

**Format:**

The exam is 150 minutes long. The exam consists of 5 questions worth a total of 50 marks. The exam booklet contains 8 pages, including the cover page and one blank page at the back of the exam booklet for rough work.

**Reference Materials:**

None.

**Instructions:**

1. Write your name and email address in the spaces above.
2. Answer each question either in the space provided or on a blank page. If you use a blank page to write your answer, indicate this clearly in the space provided for the question. Show all of your work.
3. Ensure that your exam booklet contains 8 pages. Do not detach any pages from your exam booklet.
4. Do not use any unauthorized reference materials or devices during this exam.
5. Sign in the space below. Your signature indicates that you understand and agree to these instructions and the university's examination policies.

**Signature:** ____________________

| Question | Marks | Score |
|---|---|---|
| 1 | 10 | |
| 2 | 10 | |
| 3 | 10 | |
| 4 | 10 | |
| 5 | 10 | |
| Total | 50 | |

## Multiple Choice

[10 marks] 1. For each of the following questions, select exactly one answer by circling the associated letter. Incorrect answers will not be penalized. Answers with more than one letter circled will be marked as incorrect.

(a) Which of the following is the correct definition of the competitive ratio $\alpha$?
   A. $\alpha = \text{ALG}/\text{OPT}$
   B. $\alpha = \text{OPT}/\text{ALG}$
   C. $\alpha = \text{OPT} \cdot \text{ALG}$
   D. $\alpha = 1/\text{ALG}$

(b) Which of the following online algorithms gives the best competitive ratio $\alpha$ for the paging problem?
   A. The LRU algorithm.
   B. The Marking algorithm.
   C. The Min algorithm.
   D. The Random algorithm.

(c) Which of the following statements about randomized algorithms is **true**?
   A. Randomized algorithms always produce correct output.
   B. Randomized algorithms exhibit deterministic behaviour.
   C. Randomized algorithms use expected case analysis to evaluate performance.
   D. Randomized algorithms are only used in theory and have no practical applications.

(d) Recall the notions of Las Vegas and Monte Carlo algorithms. Is it possible to convert from one type of algorithm to the other?
   A. Yes, always.
   B. Sometimes, from Monte Carlo to Las Vegas; yes, from Las Vegas to Monte Carlo.
   C. Yes, from Monte Carlo to Las Vegas; sometimes, from Las Vegas to Monte Carlo.
   D. No, never.

(e) Consider the following pseudocode, where the length-$j$ array $B$ is initialized to all ones. What is the amortized complexity of running this pseudocode $n$ times?

```
x = 0
while (x < j && B[x] == 0):
    B[x] = 1
    x = x + 1
if (x < j):
    B[x] = 0
```

   A. $O(1)$
   B. $O(\log(n))$
   C. $O(n)$
   D. $O(n \log(n))$

(f) Suppose we perform a sequence of operations such that the $i$th operation costs \$1 if $i$ is *not* a power of 2, and costs \$$i$ if $i$ is a power of 2. The following table shows the costs for the first few operations:

| **Operation** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cost** | \$1 | \$2 | \$1 | \$4 | \$1 | \$1 | \$1 | \$8 | \$1 | \$1 | ··· |

What is the best upper bound on the amortized cost per operation in this sequence?

A. \$2
B. \$3
C. \$log($n$)
D. \$$n$

(g) Consider the following subsets corresponding to an instance of the set cover problem:

$$S_1 = \{3,5,7\};\ S_2 = \{3,4,7,9\};\ S_3 = \{2,6\};\ S_4 = \{4,8,9\};\ S_5 = \{3,8\}.$$

What is the correct value of $f$ for the set cover $f$-approximation algorithm?

A. 1 (the least frequency of any element)
B. 3 (the greatest frequency of any element)
C. 9 (the largest element)
D. 2 (the smallest element)

(h) Which of the following statements about integer and linear programming is **<u>true</u>**?

A. Integer programming is not NP-hard, while linear programming is NP-hard.
B. $z^*_{\text{IP}} \leq z^*_{\text{LP}} = \text{OPT}$.
C. We can obtain an $\alpha$-approximation algorithm if we can find an integral feasible solution within a factor $\alpha$ of $z^*_{\text{LP}}$.
D. Every feasible linear program solution is also a feasible integer program solution.

(i) What is the minimum Hamming distance between any pair of strings in the following set?

$$\{\mathsf{00000}, \mathsf{01101}, \mathsf{10110}, \mathsf{11011}\}$$

A. 1
B. 2
C. 3
D. 4

(j) Which of the following statements about a problem $P$ is **<u>not</u>** equivalent to the others?

A. $P$ is a fixed-parameter tractable problem.
B. We can reduce instances of $P$ to a kernel of size $< f(k)$.
C. There exists an algorithm solving instances of $P$ in time $O(n^c + f(k))$.
D. There exists a bounded search tree of height $< n^c$ enumerating solutions to $P$.

## Short Answer

[10 marks] 2. For each of the following questions, give a 1–2 sentence answer.

(a) Why are online paging algorithms like LIFO and LFU not $\alpha$-competitive for any value of $\alpha$?

(b) What is the difference between a Monte Carlo algorithm having one-sided error and a Monte Carlo algorithm having two-sided error?

(c) How does the potential method differ from the accounting method when performing an amortized analysis?

(d) Given a linear program $L$, what three changes must we make in order to formulate the dual of $L$?

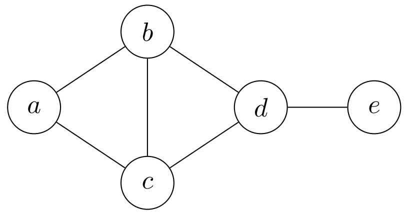
(e) What does it mean to "kernelize" a problem, and what does kernelization imply about the existence of an algorithm solving the problem?

[10 marks] 3. Suppose we want to perform a sequence of push and pop operations on a stack whose size never exceeds some integer $k \geq 1$. Additionally, after every $k$ operations, we make a copy of the entire stack as a backup. (Where we store this backup is not relevant to the question.)

Show that the amortized cost of any sequence of $n$ stack operations (i.e., pushing, popping, or copying the stack) is $O(n)$. You may use any method of amortized analysis you like.

[10 marks] 4. Recall the definition of the vertex cover problem.

(a) What is an optimal vertex cover for the following graph?




(b) Formulate an integer program modelling the vertex cover problem, and explain how you would relax it to a linear program for the same problem.

(c) Give an example solution to your relaxed linear program from part (b) that has a value strictly less than OPT, where OPT is the optimal solution size you found in part (a).

[10 marks] 5. Suppose you work as a security guard in an art gallery, and you are asked to figure out where to place some number of surveillance cameras within the gallery such that every corridor of the building is monitored. The gallery has a tight budget, though, so you must also minimize the number of surveillance cameras needed.

(a) Explain how we can formalize this art gallery problem as an optimization problem *using graphs*.

You should formulate your answer in two parts: specify the input that we are given, and specify the property that we want to solve/optimize.

(b) Let $n$ denote the number of locations in the building where two corridors intersect, and let $k$ denote the number of surveillance cameras you have to place. Instead of checking all $2^n$ possible surveillance camera setups, you can apply parameterization to this problem to make your job easier.

Describe how you can build a search tree for the art gallery problem that requires only $k$ checks to determine whether a solution exists.

[2 marks] *Bonus.* What was your favourite part of this course, and why?

This blank page may be used for rough work.