

CS-203 – FINAL EXAM – 2022 SUMMER  
GIUSEPPE TURINI – KETTERING UNIVERSITY

*Instructions*

- This exam is take-home, open-book, open-notes, and individual (no collaboration).
- Each part indicates the points awarded if correctly answered (partial credit available).
- Submit your solution as a single PDF file, via email using your Kettering account.
- The submission deadline is: Saturday 24 September 2022, before the end of the day.

*Student Information*

- Student full name (readable) and signature:

*Exercise 1 (50 points)*

Consider this algorithm designed to process a square matrix (2D array  $N \times N$ ) in input.

```
// Input: 2D square matrix A of size NxN.
public static boolean isSymmetric( int[][] A ) {
    for( int i = 0; i < A.length-1; i++ ) {
        for( int j = i+1; j < A.length; j++ ) {
            if( A[i][j] != A[j][i] ) { return false; }
        }
    }
    return true;
}
```

Analyze this algorithm and answer the following questions.

- Determine: (1) the input size, (2) if best-case and worst-case analyses are needed, and (3) the basic operations, motivating your answer (max 50+50+50 words, 5+5+5 points).
- Define: (1) the basic operation count as a summation, (2) its conversion into a closed-form expression, and (3) this algorithm efficiency class (5+5+5 points).
- Implement this algorithm recursively, specifying its category/design (e.g., decrease-by-one, etc.) and efficiency class, motivating your answer (max 150 words, 20 points).

**Note** Assume the input matrix is square (2D array  $N \times N$ ), and valid (not null).

**Note** If necessary, perform this analysis for both the best-case and worst-case.

*Exercise 2 (20 points)*

Consider a hash table HT with a capacity of 11 items, and this hash function HF.

```
// Primary hash function.
private static int HF( int key ) {
    return key % HT.length;
}
```

Answer these questions.

- a Insert these keys (left to right) in HT: 163, 177, 195, 140, 151, 136, 122, 114, 188; using linear probing and drawing the hash table HT, only once, after all the insertions above have been performed (10 points).
- b Resize the hash table HT using a new capacity of 23 items, properly rehashing its content; this time using quadratic probing and drawing the hash table HT, only once, after the resizing/rehashing is complete (10 points).
- c Compare the status of the hash table HT in the previous two answers (*e.g.*, time/space efficiency, load factor, collisions, clustering, *etc.*), determining if the resizing and new collision resolution scheme were appropriate (extra points, max 150 words, 10 points).

**Note** Question (c) awards 10 extra points (for a maximum of 110 points total).

*Exercise 3 (30 points)*

Consider these characters (from left to right): “C”, “O”, “P”, “Y”, “R”, “I”, “G”, “H”, “T”.

- a Construct an AVL tree by inserting the characters above one by one: (1) showing the AVL tree after each character insertion, and (2) listing each rotation performed with its type and the pivot nodes (15 points).
- b Construct a 2-3 tree by inserting the characters above one by one: (1) showing the 2-3 tree after each character insertion, and (2) including all temporary 4 nodes created and their necessary splits (15 points).

**Note** Use the alphabetical order of letters to sort characters.