ROYAL HOLLOWAY, UNIVERSITY OF LONDON
BSc EXAMINATION 2021

CS2860: Algorithms and Complexity

CS2860R: Algorithms and Complexity – FOR FIRST SITS/RESIT CANDIDATES

Time allowed: **TWO hours**

Please answer **ALL** questions.

- Handwrite your answers on paper, and write your candidate number and the module number at the top of each page. Photograph/scan the pages and keep the original paper versions, as they may be required by the examiners.
- For each question you attempt, please clearly state the question number.
- Please DO NOT include your name or Student ID anywhere on your work.
- **Academic Misconduct:** We will check all assignments for academic misconduct. Suspected offences will be dealt with under the College's formal Academic Misconduct procedures. Please remember:
  - The work submitted is expected to be your own work and only your work. You may not ask for help from any source, or copy anyone else's work.
  - You must not give help to anyone else, including sending them any parts of the questions or copies of your solutions.
  - You must not discuss the questions or solutions with anyone else.
- **Submitting your work:**
  - Your document must be submitted through Moodle using the submission link in the module Moodle page. If possible please convert your document into a PDF document to make the submission process quicker and easier.
  - Emailed submissions will not be accepted.
  - **You must complete your exam upload within 1 hour of the exam finish time.**

1. (a) Describe the order of growth of the following functions in standard Θ notation as simply as possible. Sort the resulting Θ expressions by increasing speed of growth. [10 marks]

   i. $f(n) = 7(\log n)^4 + 3n + 5$

   ii. $f(n) = 3n^2 + 4n \log n + 42$

   iii. $f(n) = n \log n - 17 \log n + \frac{7n-2}{\log n}$

   iv. $f(n) = 42 \log n - 69$

   (b) Explain what is the meaning of c and n_0 in the definition of Big-Oh notation. Given $3n^3 - n = O(n^4)$, find the minimum possible value of c for $n_0 = 1$ and for $n_0 = 10$. [6 marks]

2. Consider the functions `fnA` and `fnB` listed below. Compute an estimate of the execution time of each function as a function of n. Give your answer in Big-Oh notation and justify your answer.

```
int fnA(int n) {
    int sum=0;
    for (int i=0; i<n; i++) {
        int j = 1;
        while ( j < i) {
            sum += i + 2j;
            j = j*2;
        }
    }
    return sum;
}

int fnB(int n) {
    int sum=0;
    for (int i = 1; i<n; i=2*i) {
        for (int j = 1; j<i; j = j*2) {
            for (int k = 0; k<j; k++) {
                sum += i+j+k;
            }
        }
    }
    return sum;
}
```

[12 marks]

NEXT PAGE

3. Show the execution of the `quicksort` sorting algorithm on the array `A=[9, 19, 8, 20, 1, 4, 17, 13, 7, 12]`. Assume an implementation that uses the first element as pivot. Illustrate all recursive calls made by the algorithm by drawing the recursion tree. [8 marks]

4. Consider the binary search tree (BST) illustrated in figure 1 below.

   (a) Define the *balance* property of AVL trees. Which nodes in the tree in figure 1 do not have the balance property? [4 marks]

   (b) Perform the following operations on the binary search tree (BST) in the given order, and show the shape of the tree after each one: Delete 20. Delete 47. Insert 25. Delete 42. Show the ordinary binary search tree operations; do not rebalance the tree after insertions or deletions. [4 marks]

   (c) What is the purpose of maintaining the balance property for AVL trees? [4 marks]

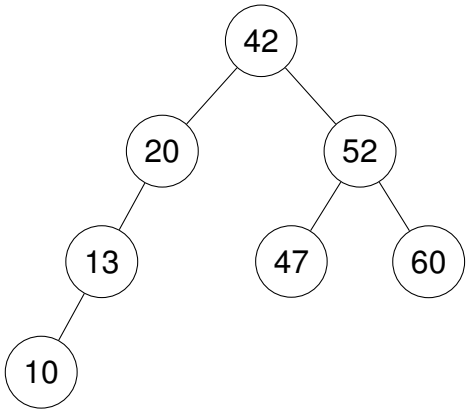

FIGURE 1: BST FOR QUESTION 4.

5. Consider the AVL tree illustrated in figure 2 below. Insert 25 into the tree, and show the steps carried out to maintain the balance property. [10 marks]

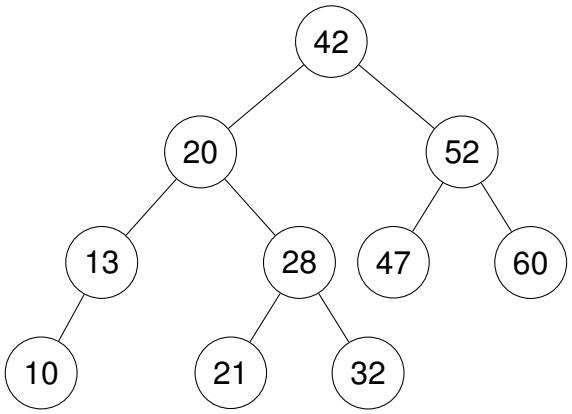

FIGURE 2: BST FOR QUESTION 5.

6. (a) State the heap property for binary min-heaps. [2 marks]

   (b) Which of the following two arrays is a binary min-heap?

   i. `A=[13, 16, 76, 26, 20, 71, 96, 36, 79, 21]`

   ii. `B=[25, 38, 30, 41, 51, 74, 92, 44, 47, 64]` [2 marks]

   (c) Consider the following array `C = [8, 5, 2, 6, 3, 1, 4, 9]`.

   i. Describe the `makeHeap` algorithm from lectures and show the steps of the algorithm applied to the array `C` above. [6 marks]

   ii. Draw the tree representation of the final binary min-heap. [2 marks]

7. Consider inserting the keys 22, 14, 76, 24, 58, 86, 74, 7, 32, 39 into a hash table of length $m = 13$ using **open addressing** with the auxiliary hash function $h_0(k) = k$ (that is, the hash of a key is equal to the key). Illustrate the result of inserting these keys using **linear probing**. [10 marks]

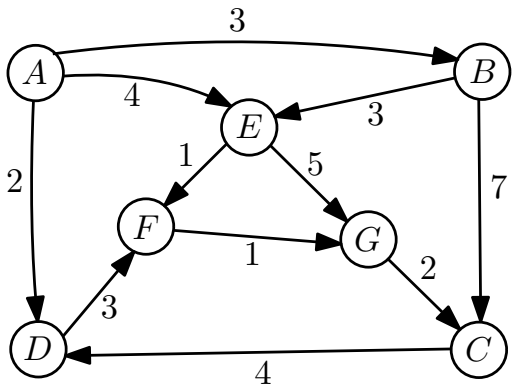

FIGURE 3: DIRECTED GRAPH FOR QUESTION 8.

8. (a) Describe Dijkstra's algorithm for finding single-source shortest paths in a graph, and illustrate it on the graph of Figure 3, by finding shortest paths to all destinations starting from vertex A. [7 marks]

   (b) Draw the resulting shortest paths tree. [2 marks]

   (c) Explain how to get an efficient implementation of Dijkstra's algorithm, i.e., an implementation that runs in time $O(m \log n)$ or better for a graph with m edges and n vertices. Justify your claim about the running time. [2 marks]

NEXT PAGE

9. You wish to check for plagiarism in a large collection of student essays. Assume that the essays are provided in plain text format, perhaps as String objects, or as arrays of char. In order to detect plagiarism, you wish to list all cases where identical substrings of length 150 characters occur in two or more essays.

   The algorithm should scan through the essays and at the end it should output a list of lists of positions in different essays where identical substrings occur. This list will then be used for a further stage of plagiarism checking; you need not consider this further stage of checking, but your algorithm design can assume that a further stage of checking will take place.

   Outline your approach with an algorithm sketch (you need not give code), and state which data structures you will use and any important aspects of the implementation.

   Also state in Big-Oh notation what the execution time should be for your algorithm, if there are n essays of length at most m characters each. You should try to make your algorithm as efficient in both time and storage as you can. [9 marks]

END