Durham University

Examination Paper

Examination Session:	Year:	Exam Code:
May/June	2020	COMP1081-WE01

Title: Algorithms and Data Structures

Time Allowed:	2 hours	
Additional Material provided:		
Materials Permitted:		
Calculators Permitted:	Yes	Models Permitted: Casio FX-83 GTPLUS, Casio FX-85GTPLUS, Casio FX83-GTX or Casio FX85-GTX
Visiting Students may use dictionaries: Yes		

Instructions to Candidates:	Answer ALL questions. Please answer each question in a separate answer booklet.

Revision:	

Section A Prof. Matthew Johnson

Question 1

(a) A linked list is made up of nodes. What are the two parts of a node? **[2 Marks]**

(b) A linked list could be used to implement a stack. The top of the stack would be at the head of the list. For example, here is the **push** method.

---

push(e)

```
node N
N.data = e
N.next = L.head
L.head = N
if L.size = 0 then
  L.tail = N
end if
L.size = L.size + 1
```

Write pseudocode for the **isEmpty** and **pop** methods for a stack implemented with a linked list. (You can assume that L.size is defined.) **[9 Marks]**

(c) Consider the function **L**(n) whose input is a positive integer.

L(n)

```
if n = 1 then
  return 0
else
  return 1+ L(n//2)
end if
```

Note that $n//2$ denotes the integer part of $n/2$. What is the output of **L**(n)? Justify your answer. **[4 Marks]**

this question is continued on the next page

(d) Suppose that you are given an array $A = [a_1, a_2, \ldots, a_n]$ of n positive integers, and a special integer m which appears at least once in the array. A subarray of A is a sequence of integers that appear consecutively in A and the weight of a subarray is the sum of its numbers. For example if A is $[1, 3, 2, 6, 2, 5]$ then $[1, 3, 2]$ and $[2, 6, 2]$ are subarrays with weights of 6 and 10 respectively.

Design an algorithm that finds the maximum weight of a subarray of A such that the subarray contains m exactly once. For example, for the array $[1, 3, 2, 6, 2, 5]$ with $m = 2$, the algorithm should return 13 (the weight of the subarray $[6, 2, 5]$). The algorithm should read each integer in the array only once.

Write pseudocode for your algorithm. **[10 Marks]**

Section B Dr. Tom Friedetzky

Question 2

(a) Define the asymptotic class o. **[2 Marks]**

(b) Let α, β be constants with $1 < \alpha < \beta$. Let $f(n) = \alpha^n \cdot \log_2(n)$ and $g(n) = \beta^n$. Prove or disprove $f(n) = o(g(n))$. **[3 Marks]**

(c) Consider functions $f, g, h : \mathbb{N} \to \mathbb{R}^+$. Is it true that $f(n) \cdot g(n) = \Theta(h(n))$ implies $f(n) = O(h(n))$ and $g(n) = O(h(n))$? Prove or disprove your claim. **[5 Marks]**

(d) Sketch a proof for the worst-case running time of generic QuickSort (i.e., one pivot element from fixed position, say, always rightmost) being $\Theta(n^2)$. **[5 Marks]**

(e) Consider the following Python function:

```
def foo(seq):
  i = 0
  while i < len(seq):
    if i == 0 or seq[i-1] <= seq[i]:
      i = i + 1
    else:
      seq[i], seq[i-1] = seq[i-1], seq[i]
      i = i - 1
  return seq
```

Explain the purpose of the function, justify its correctness and provide asymptotically tight running time bounds. **[10 Marks]**

End of Section B

Section C Prof. Andrei Krokhin

Question 3

(a) Describe the selection problem and the Median-of-Medians algorithm for it. Your description of the algorithm can be given as an itemised list in plain English. There is no need to give pseudocode. **[7 Marks]**

(b) Manually run the HeapSort algorithm on the following array: $[1, 9, 6, 8, 5]$. Show and explain your working. **[9 Marks]**

(c) i. Is Median-of-Medians an optimal selection algorithm (in some specific sense)? **[3 Marks]**

ii. Is HeapSort an optimal sorting algorithm (in some specific sense)? **[3 Marks]**

Justify your answers. In your justifications, you can refer to statements proved in the lectures — there is no need to reprove them.

(d) Prove that any correct comparison-based algorithm to find the smallest element in an unsorted array (of integers) of length n must make at least $n - 1$ comparisons. **[3 Marks]**

Section D Prof. Matthew Johnson

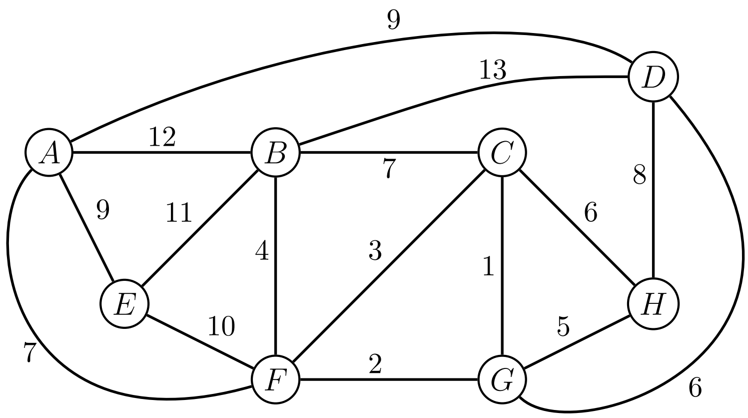
Question 4

(a) From a depth-first search of a directed graph G, a depth-first forest F is obtained that contains the edges traversed during the search. The edges of G can be classified with respect to F. Explain the properties of an edge classified as a cross edge. **[2 Marks]**

(b) Explain why, if we use the same classification of edges for an undirected graph G and a depth-first tree T, no edge will be classified as a cross edge. **[3 Marks]**

(c) From a breadth-first search of a connected undirected graph G, a breadth-first tree T is obtained that contains the edges traversed during the search. An edge e joining two vertices u and v in G is called a bridge if the deletion of e creates a graph that is not connected. Is the following statement true or false: an edge in G is a bridge if and only if it is included in every possible breadth-first tree of G? Justify your answer. **[9 Marks]**

(d) Consider the graph below. List the edges of a minimum spanning tree (MST) in the order they are found when Prim's algorithm is applied starting at vertex A.



[3 Marks]

this question is continued on the next page

(e) Let T be an MST of a weighted undirected graph G. Let (u, v) be an edge of T, and let T_1 and T_2 be the two trees obtained if (u, v) is removed from T. Let G_1 be the subgraph of G that contains the vertices of T_1 and all the edges of G between those vertices.

Show that T_1 is an MST for G_1, and that (u, v) is a least weight edge from a vertex of T_1 to a vertex of T_2. **[4 Marks]**

(f) Consider the following algorithm for finding an MST of a graph G.

- Divide the vertices of G into two roughly equal sets V_1 and V_2.
- Let G_1 be the subgraph of G that contains the vertices of V_1 and all the edges of G between those vertices.
- Let G_2 be the subgraph of G that contains the vertices of V_2 and all the edges of G between those vertices.
- Find MSTs of G_1 and G_2. Let these two trees be T_1 and T_2.
- Find a least weight edge (u, v) that joins a vertex in V_1 to a vertex in V_2.
- The output of the algorithm is the tree formed from T_1, T_2 and (u, v).

Is the algorithm correct? Justify your answer. **[4 Marks]**

END OF PAPER