Section A Part I
(Prof Matthew Johnson)

Question 1

(a) What does an initially empty stack contain after the following sequence of operations?

> push(5), push(9), pop, push(1), push(6), pop, push(4), pop, pop, push(7).

Moreover, what does the operation top return if applied after the sequence above? **[2 Marks]**

(b) Suppose S_1 is a non-empty stack. Design an algorithm whose input is S_1 and whose output is S_1 in its initial state and a second stack S_2 that is a copy of S_1.

The algorithm can use a single queue Q and can store the value of a single element in a temporary variable, but no other data structures can be used, and only the operations push and pop (on the stacks), and enqueue and dequeue (on the queue) can be used.

You can describe the algorithm in a short paragraph or write pseudocode. **[8 Marks]**

this question is continued on the next page

(c) Consider the pseudocode below that describes an algorithm whose input is a linked list where each piece of data is an integer. Describe the output, and explain the roles of x, y and z.

Input: non-empty linked list L containing integers
Output: ???

```
x = 0
y = L.head
z = d.next
while z ≠ NULL do
    if z.data < y.data then
        x = x + 1
    end if
    y = z
    z = y.next
end while
return x
```

[7 Marks]

(d) Consider the Trajan numbers T_n ($n \geq 0$ integer) defined by

$$T_n = \begin{cases} 5n & \text{for } n \leq 4, \\ T_{n-1} - T_{n-3} + 3T_{n-5} & \text{for } n \geq 5. \end{cases}$$

i. Calculate T_{10}. **[1 Mark]**

ii. Write pseudocode for a **non-recursive** function that returns T_n for an integer $n \geq 0$. **[3 Marks]**

iii. Write pseudocode for a **recursive** function that returns T_n for an integer $n \geq 0$. **[3 Marks]**

iv. How could you improve the running time of the recursive function?

[1 Mark]

Section B Part II
(Dr Konrad Dabrowski)

Question 2

(a) In the QuickSort algorithm, suppose you have a partition function that:

- always chooses the second-smallest element as the pivot,
- keeps the elements less than the pivot in the same order with respect to each other and
- keeps the elements greater than the pivot in the same order with respect to each other.

i. Apply the QuickSort algorithm from lectures, but with this partition function instead, to sort the following array:

$$1, 83, 74, 26, 63, 37, 25$$

Show the state after every time the partition function is called.

[5 Marks]

ii. Suppose the partition function above runs in $\Theta(n^2)$ time. Give tight upper and lower bounds for the runtime of QuickSort with this partition function (you may assume that n is odd and that no two elements of the input array are equal). Justify your answer. **[5 Marks]**

(b) Consider functions $f, g : \mathbb{N} \to \mathbb{R}^+$. Is it true that $f(n) = \Theta(g(n))$ implies that $3^{f(n)} = O(4^{g(n)})$? Prove or disprove your claim. **[5 Marks]**

this question is continued on the next page

(c) The input for algorithm D is an array A of positive integers $A[1], ..., A[n]$.

```
D(A)
n = length(A)
for i=2 to n do
  y = A[i]
  j = i-1
  while j>0 and (A[j]<y or A[j] is even) do
    A[j+1]=A[j]
    j=j-1
  end while
  A[j+1]=y
end for
return A
```

Describe what this algorithm does. Find tight upper and lower bounds on the asymptotic runtime of this algorithm (i.e. find the best possible $f(n)$ and $g(n)$ such that the algorithm always uses $O(f(n))$ comparisons and $\Omega(g(n))$ comparisons). Justify your answers. **[10 Marks]**

Section C Part III
(Prof Andrei Krokhin)

Question 3

(a) Consider the problem of selecting the i-th smallest element and the QuickSelect algorithm which uses the partition function that

- always chooses the leftmost element of the subarray $A[\text{left} \ldots \text{right}]$ as the pivot,
- keeps the elements less than the pivot in the same order with respect to each other and
- keeps the elements greater than the pivot in the same order with respect to each other.

Manually run the QuickSelect algorithm with the above rule on the following input:

$$i = 3 \text{ and } A = [13, 7, 5, 2, 4, 12, 8, 10, 1].$$

[5 Marks]

(b) Suppose we have integer values between 1 and 1000 in a binary search tree and search for 437. Which of the following cannot be the sequence of keys examined? Explain your answer. **[5 Marks]**

i. 7, 8, 9, 987, 654, 533, 426, 502, 437

ii. 523, 466, 123, 201, 455, 453, 202, 302, 437

iii. 89, 859, 158, 700, 233, 699, 249, 512, 437

iv. 700, 120, 612, 523, 258, 302, 345, 580, 437

(c) Construct an AVL tree T and 8 numbers $a_1, a_2, \ldots, a_8$ such that when inserting these numbers into T, in this order, a rotation is performed during the fix-up procedure after each insertion. **[8 Marks]**

(d) For an array A of pairwise distinct integers, let $Heap(A)$ denote the max-heap obtained by applying the algorithm BuildHeap to A. For example, if $A = [1, 2, 3, 4]$ then $Heap(A) = [4, 2, 3, 1]$. Construct an array A consisting of distinct positive integers such that the numbers $1, 2, 3, 4$ appear in A in this order (not necessarily consecutively), but they appear in $Heap(A)$ in the opposite order, i.e. $4, 3, 2, 1$ (also not necessarily consecutively). **[7 Marks]**

End of Section C

Section D Part IV
(Dr George Mertzios)

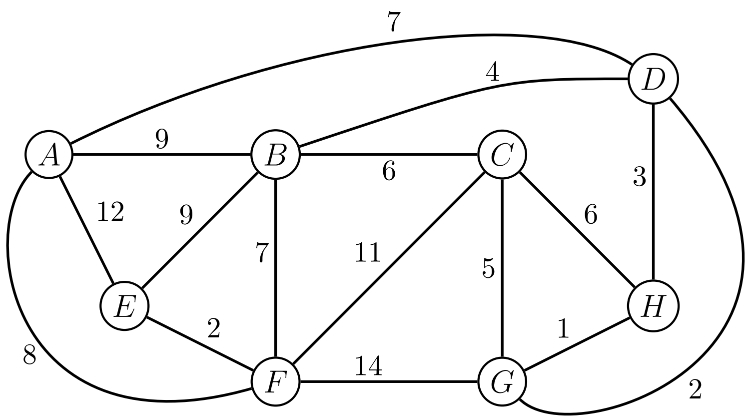
Question 4

(a) Suppose a breadth-first search is run on the directed graph given by the adjacency lists below with vertex 1 being the source. Classify each edge of the graph as a tree, a forward, a back or a cross edge.

$$\begin{array}{rl} 1: & 3,4 \\ 2: & 1 \\ 3: & 7 \\ 4: & 8,2 \\ 5: & 4 \\ 6: & 4,8 \\ 7: & 6,8 \\ 8: & 5,9 \\ 9: & - \end{array}$$

[8 Marks]

(b) Suppose that Prim's algorithm is applied to the graph below, starting at vertex A. In how many different sequences can the edges of the resulting minimum spanning tree be discovered? Justify your answer. List the edges of a minimum spanning tree in the order they are found, according to one of those correct edge sequences.



[5 Marks]

this question is continued on the next page

(c) Suppose that we have an undirected graph with weights on the edges, where each weight is either a positive odd number or a negative even number. Does Kruskal's algorithm produce a minimum spanning tree in such a graph? Justify your answer. **[5 Marks]**

(d) Let G be a connected undirected weighted graph, and suppose that G contains at least one cycle C. Let $e = (u, v)$ be an edge of the cycle C which has the greatest weight in the cycle. Show that there exists a minimum spanning tree of G that does not contain edge e. **[7 Marks]**

END OF PAPER