

- This exam has 3 questions for a total of 150 marks, of which you can score at most 100 marks.
- Use extra sheets to do your rough work. Minimize rough work on the answer sheets.
- You will **not** get the credit for an answer if you do one of the following:
 - Not clearly state all results that you use;
 - Use arguments with implicit assumptions;
 - Use unclear or incomplete reasoning, or appeals to intuition;
 - Use “weasel words” such as “clearly” and “obviously”.
- It is OK to cite results that were stated and/or proven in class, and (parts of) existing answers from your own answer sheet. You may also freely use well-known properties of mathematical objects, but you must clearly state what it is that you use.
- You will get partial credit for answers which are partly correct. So you should consider writing down what you know to be correct, even in those cases where you cannot provide a proof.

1. Using loop invariants, prove that given an array A of $n \geq 2$ integers as input, procedure INSERTION SORT described in Algorithm 1 returns an array B such that:

[50]

1. B is a permutation of the elements of A , and
2. $B[i] \leq B[i + 1]$ holds for all values $i \in \{1, 2, \dots, (n - 1)\}$.

Algorithm 1 Insertion Sort. A is an array of integers, index starting at 1. A has at least two elements. $A[i]$ is the i^{th} element of A . $\text{length}[A]$ is the number of elements in A .

1: procedure INSERTION-SORT(A)

2: $i \leftarrow 2$

3: $n \leftarrow \text{length}[A]$

4: while $i \leq n$ do

▷ Loop 1 starts here.

5: $\text{next} \leftarrow A[i]$

6: $j \leftarrow (i - 1)$

7: while $j > 0$ and $A[j] > \text{next}$ do

▷ Loop 2 starts here.

8: $A[j + 1] \leftarrow A[j]$

9: $j \leftarrow (j - 1)$

10: end while

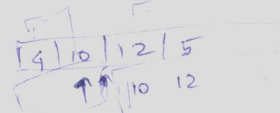
11: $A[j + 1] \leftarrow \text{next}$

12: $i \leftarrow i + 1$

13: end while

14: return A

15: end procedure



2. The *adjacency matrix* of a simple undirected graph $G = (V = \{v_1, \dots, v_n\}, E)$ is an $n \times n$ matrix Adj such that $\text{Adj}[i, j]$ is 1 if $v_i v_j$ is an edge in G , and is 0 otherwise. Vertex v_i is said to be *connected* to vertex v_j in G if there is a path from v_i to v_j in G . A vertex is connected to itself. A *connected component* (or *component* in short) of G is an equivalence class of the "is connected to" relation. Graph G is said to be *connected* if it has exactly one connected component, and is *disconnected* otherwise.

You should *not* provide the pseudocode or correctness argument of BFS in your solution for any of the following questions.

- (a) Describe the input and output of the BFS algorithm. (This is called the "black box" behaviour of BFS.) [5]

For each of the next three questions your algorithm should use BFS as a black box in the manner that you specified in your solution to part (a), and should return YES, NO, or a number, as appropriate. You need *not* give a proof of correctness of your algorithm.

- (b) Describe an algorithm which, given vertices u, v of G , checks if there is a path from u to v in graph G . [5]
 (c) Describe an algorithm which checks if graph G is connected. [5]
 (d) Describe an algorithm which finds the number of connected components of graph G . [5]

For each question below, one input is the adjacency matrix Adj of graph G . For each of these questions your algorithm should NOT use BFS or DFS¹, as a black box or otherwise. It should return YES or NO, as appropriate. In each case you **should prove** that your algorithm is correct, and derive a good asymptotic upper bound on its running time.

- (e) Describe an algorithm which, given vertices u, v of G , checks if there is a path from u to v in G . [15]
 (f) Describe an algorithm which checks if graph G is connected. [15]
3. Give asymptotic upper and lower bounds for $T(n)$ for each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n . Make your bounds as tight as possible, and justify your answers. *Should*
- (a) $T(n) = T(n-2) + n^2$ [10]
 (b) $T(n) = T(n-1) + \frac{1}{n}$ *1/2* [10]
 (c) $T(n) = T(n-1) + \lg n$ *✓* [10]
 (d) $T(n) = T(n-2) + \frac{1}{\lg n}$ *α* [10]
 (e) $T(n) = \sqrt{n}T(\sqrt{n}) + n$ *1/2* [10]

¹In the form described in class.