# PG4200 Algorithms and Data Structures

# Re-sit exam

## 6/8/2025

Candidate: 31

### Question 6:

#### LO5 and LO6: Computability and Complexity

#### 1. What is a Complexity Class? Why is it known as complexity classen in data structures?

A complexity class is a group of computional problems that require similar amounts of time and/or memory to solve. They are called «complexity classes» because problems are organised into sets based on how difficult they are to compute (Gupta, n.d.; Goodrich & Tamassia, 2006, Ch.14; Sedgewick & Wayne, 2011, Ch.6).

In datastructures, these classes help us choose the right algorithms depending on efficiency needs.

##### Real-life example:

Sorting letters in a post office. If you have 10 letters, you can sort them quickly; with 10 000 letters, it takes longer. The sorting method you use (by postcode, aplhabetical, automated) changes the complexity class of the task.

#### 2. Why do we use Big O Notation to analyse algorithm´s complexity in data structures?

We use Big O notation to describe the upper bound of an algorithm´s running time or memory use as the input size grows. It abstracts away hardware details and focuses on scalability (Sedgewick & Wayne, 2011, Ch.1; Goodrich & Tamassia, 2006, Ch.5).

##### Real-life example:

When loading a website, some designs scale poorly, adding more images and scripts can drastically slow it down. Big O helps predict hos performance changes with size, letting developers optimise the «algorithm» behind page rendering.

#### 3. What is the P Class?

The P class contains problems that can be solved in polynomial time by a deterministic computer (Goupta, n.d.; Goodrich & Tamassia, 2006, Ch.14).

Features:

* Solvable in a reasonible time, even for large inputs.
* Deterministic, same input always gives the same output quickly.
* Considered tractable (practical to solve).

##### Real-life example:

Finding the fastest route on Google Maps is in P, even with thousands of roads, algorithms like Dijkstra´s can compute it in milliseconds. Sorting a playlist by song length is another P-class problem.

#### 4. What is the NP class?

NP problems have solutions that can be verified in polynomial time, even if finding the solution may take much longer (Gupta, n.d.; Goodrich & Tamassia, 2006, Ch.14; Sedgewick & Wayne, 2011, Ch.6).

Features:

* Solution us easy to check, hard to find.
* Includes all the problems in P.
* Often requires brute-force search if no faster algorithm is known.

##### Real-life example:

Solving a jigsaw puzzle, you can quickly check if a finished puzzle is correct, but figuring out where every piece goes can be time-consuming. Another example is arranging studenst in an exam hall so no friends sit together, easy to check, hard to plan.

#### 5. What is an NP-complete class?

NP-complete problems are the hardest problems in NP, they are both in NP and NP-hard (Gupta, n.d.; Goodrich & Tamassia, 2006, Ch.14; Sedgewick & Wayne, 2011, Ch.6).

Features:

* Any NP problem can be reduced to an NP-complete problem in polynomial time.
* If one NP-complete problem is solved in polynomial time, all NP problems an be solved in polynomial time.
* No known efficient algorithm exists to solve them exactly.

##### Real-life example:

The Travelling Salesman Problem, finding the shortest route visiting all delivery stops once is very hard to compute, but easy to check once a route is given. A sports tournament schedule that minimises travel time is another NP-complete problem.

### Referances

Goodrich, M. T., & Tamassia, R. (2006). *Data Structures and Algorithms in Java* (4thed.). Wiley.

Sedgewick, R., & Wayne, K. (2011*). Algorithms* (4thed.). Addison-Wesley.

Gupta, R. (n.d) *LO 6: Computability and Complexity* (Lecture slides). Kristiania University College.