

# Second Year Tutorial (Semester 2): COMP26120

## Prepare a presentation on Knapsack Problems

This tutorial sheet is **essential preparation** for

- a laboratory exercise for COMP26120. This is a large three session exercise, running in weeks 5,6 and 7.
- the summer examination which will cover this material.

You are asked to prepare a joint presentation in your tutorial group on **Knapsack problems**, their applications and their solutions.

To prepare your presentation, you should **arrange to meet as a group outside the tutorial time** in weeks 3 or 4 to decide how to split the topics up amongst yourselves (making sure that all are involved) and how the topics should be presented. You should prepare for this tutorial well ahead of the tutorial time and before the laboratory session starts.

**Warning:** Each topic should be covered in approximately 10 minutes. You will not have much time to talk on technical aspects, so should give a good overview of each topic.

For topics 2, 3 and 4 below, you should show how the relevant algorithm runs on the following example of a Knapsack problem:

---

There are 4 items of size/weight 2, 1, 6 and 5. The items have benefit/profit/value 10, 7, 25, 24 respectively and the maximum combined size/weight is 7 (the capacity of the knapsack). You are to choose items to fit into the knapsack with optimal (largest) combined benefit/profit/value.

---

### Topics

1. **Knapsack problems, their difficulty and applications.** Outline the definition of the problem, various forms including the *0-1 problem* and the *fractional problem*. Applications of Knapsack problems occur in transportation, logistics, cutting and packing, telecommunications, advertisements, investment, budget allocation, and production management. Give examples of knapsack problem applications in two or three of these areas. The course textbook defines the problems (as does the Wikipedia page). Applications may be found by searching on the web for ‘knapsack problems’ with ‘cutting and packing’, or ‘advertisements’, or ‘binpacking’ etc.
2. **The Fractional Knapsack problem and its greedy solution.** Explain what a ‘greedy algorithm’ is and illustrate using the example above. This is covered in Section 5.1.1 in the course textbook. Is there a greedy solution to the 0-1 problem?

3. **Simple exhaustive search for the 0-1 problem:** Solving the 0-1 Knapsack problem by exhaustive search through the space of possible solutions: To do this, we may *enumerate* all possible subsets of the items and then process these subsets (how?), or we may construct a *search tree* and use a backtracking technique. The latter is covered in Section 13.5.1 of the course textbook. Illustrate on the example above.
4. **Branch-and-bound techniques for the 0-1 problem.** The Branch-and-Bound technique - explain in general how this works and how you may use the fractional case to provide a bound. Illustrate how to do this using the example above. This is covered in Section 13.5.2 in the course textbook.
5. **Dynamic programming solutions to the 0-1 Knapsack problem.** This is covered in Section 5.3.3 in the course textbook. Explain what ‘dynamic programming’ is and show how it may be applied to the above example.

If you have time at the end, compare the time complexities of the different algorithms (a) on the example above, (b) in general.