

Practice 3 – Dynamic Programming: A well-made decision, a well-chosen treasure

Data Structures and Algorithms II. University of Almeria

Version curso.2022

Goals

- Construct solutions to a problem using the **dynamic programming algorithmic method**. Analyze and compare the implemented algorithms from different perspectives.
- Carry out an **analysis of the efficiency** of the solutions provided, and a comparison both from a theoretical and practical point of view.

Requirements

To pass this practice, the following must be done:

- Remember (from EDA I) the proper **data structures** for problem solving.
- Know how to implement the **dynamic programming scheme** to respond to a specific need (in this case, solving an optimization problem).
- **Evaluate the implemented dynamic programming algorithms**, and compare them with greedy algorithms if necessary, relating both algorithmic methods.

Problem statement

In EDALand there are very varied characters, among them, we find 'Pedrito the curious', an intrepid adventurer from Almería who, from childhood, wanted to emulate Indiana Jones, devouring books in the library about unsolved mysteries, legends of ghost towns and lost treasures. After the last major earthquake in EDALand with a magnitude of 7.3 on the Richter scale, which was felt even in its neighboring country InSoland, part of the country's orography was modified and large gaps in the southeastern mountain range were opened. This fact triggered the adventurous spark of Pedrito, who, guided by his knowledge of hidden treasures and his courage when undertaking caving challenges, has enrolled on the difficult mission of exploring the Zorokotroko cave, the highest peak in that mountain range, and where the legend tells that the great treasure of the pirate Zrokk the Bloodthirsty is hidden.

The mission is quite complicated and risky since, over time, several specialized groups

have tried to explore that cave without success, since beyond 500 meters from the entrance there is a dangerous fall of more than 1000 meters, whose end seems to be in the Center of the Earth. Pedrito, who is not afraid of anything, embarks on a new adventure, and dressed in his caving equipment, he enters the cave with the aim of finding the treasure. He does not follow the path marked by the previous groups that explored the cave, but he diverges by a totally new, never explored crack opened by the earthquake. After walking several hundred meters down an abrupt tunnel no more than a meter high, Pedrito discovers to his surprise that, when he focuses his powerful flashlight, something shining can be seen in the distance. He continues descending a few tens meters more, and his first suspicions come true when he discovers a small cavity (cave) where several objects that shine when he projects light on them, giving the sensation that they may be of great value. Pedrito's first feeling is that... he found Zrokk's treasure, EUREKA!!

When Pedrito finally manages to reach the place where all the treasure is, he realizes that, unfortunately, there are not too many objects there, but he can appreciate that they are all very valuable. Given his great knowledge of treasures and by weighing all the objects, he can have a very good estimate of the weight and value of each object that make up Zrokk's treasure.

Due to how dangerous the way back is (there are noises in the rocks advising that a landslide could occur at any moment) and the limited capacity of his caving backpack, Pedrito is faced with the difficult task of carrying only those objects that fit into his backpack, which has a limited capacity (weight it supports).



Figure 1. Pedrito's backpack and the Zrokk's treasure.

Work to develop

You must propose and implement algorithms solutions using dynamic programming to the following problems:

- Assuming that the objects cannot be broken and that the weights are exact (positive natural numbers), determine which objects Pedrito should choose to maximize the total value of what he can carry in his backpack, knowing that the values of the objects that Pedrito estimates are positive real numbers.
- Assuming that next to that cave, Pedrito discovers that there is another much larger one, full of objects that he cannot count, but he can see that there is an inexhaustible amount of precious objects of different types (classes). Each object is indivisible (it cannot be broken), it has a certain weight (positive natural) and a certain value (positive real) that Pedrito knows perfectly well. Determine how many objects of each type Pedrito should take to maximize the total value he can carry in his backpack.
- Assuming that the objects cannot be broken, but the weights may not be exact (the weights may be positive real numbers). Indicate how the algorithm of the first section should be modified so that Pedrito could carry to maximize the value in his backpack.
- Imagine the case that the valuable objects that Pedro finds in the cave can be broken (fractionated), would it be easy to find a solution with *dynamic programming* so that Pedro determines which objects to choose to maximize the total value he can carry in his backpack?. Would it be easier to do it with a **greedy** algorithm? If so, implement it and expose the main differences between the two algorithmic schemes (dynamic programming and greedy) for this specific case.

To do this, you must complete the following sections:

- **Study of the implementation:** Explain the most important details of the implementation, both of the data structures used to store the network, and of the implemented greedy algorithms. The code must be reasonably well documented (JavaDoc).
- **Theoretical study:** Study the execution times of the implemented dynamic programming algorithms depending on the number of objects and their values. Also compare the proposed algorithms, taking into account the characteristics of the specific problem case they solve and how the tables used for that aim are updated. Compare the two techniques used, dynamic programming and greedy, highlighting their characteristics, advantages, disadvantages, etc.
- **Experimental study:** Validation of the dynamic programming algorithms implemented can be done using the datasets available in the following website: https://people.sc.fsu.edu/~jburkardt/datasets/knapsack_01/knapsack_01.html. We can see that there are several datasets, with data related to: the backpack (knapsack) capacity, the weights of the objects, the profit or benefit of objects, and the optimal selection of weights. To do this, the proper operation should be checked, comparing the obtained result with the expected one, and the execution times of the implemented algorithms must be obtained and compared. The theoretical and experimental results

will be contrasted, checking if the experimental ones confirm the previously analyzed theoretical ones. The experiments carried out will be justified, and in case of discrepancy between the theory and the experiments, an attempt should be made to find a reasoned explanation. In addition, **random datasets** will be generated in the same format to the ones available on the website, for the cases of different types/classes of objects (inexhaustible number of objects in each type/class) and for the case that the objects can be divided (in this case, take into account some dataset of the website). For this last case, the data will be also tested with the implemented greedy algorithm.

Submissions

A [private](#) GitHub repository (same repository for all EDA II practices) with all the documentation and source code required in the practice must be submitted on date:

- In that repository, create a new folder called **practica_3**, where you create two subfolders, one for the documentation, **docs**, and one for the source code, **sources**.
- A memory document that explains everything you have done in practice. The memory must have the format indicated below. If desired, you can also make a presentation of the practice.
- Source code of the application, developed in JAVA, which solves everything raised in practice. Remember that you will have to measure execution times of your solutions, so you must include the necessary commands for this in the source code.
- Test suite with unit tests that you consider appropriate to make sure that everything works properly.

The **memory** of this practice to deliver must be brief, clear and well written. This should include the following sections:

- A brief **introduction** with a theoretical study of the algorithmic method used in this practice (dynamic programming).
- A section for each of the **proposed sections** to be developed in this practice (implementation study, theoretical study and experimental study). We must emphasize that the sections must be included in the same order in which they have been presented.
- An **annex** with the design of the implemented code will also be included, with a class diagram and any other diagram you consider useful, but do not include code here. In this annex, a list of the source files and a brief description of the content of each one must be included as well.
- It is important to always include the **bibliographical sources** used (web, books, articles, etc.) and refer to them in the document.

Assessment

Each section will be evaluated independently, although it is a necessary condition to pass the internship that the implemented programs work correctly.

- The implementation together with the documentation of the code will be valued out of 40%
- The study of the implementation will be valued out of 10%
- The theoretical study will be valued out of 15%
- The experimental study will be valued out of 35%

It will be penalized not deliver the theoretical introduction section or a bad presentation of the report.

The defense of the code and memory by the teacher may be required.

Deadline

Deadline: **May 08th 2022**