How do you save a video game startup just by knowing how to sort efficiently?

Problem-based learning (PBL) for students of the Advanced Algorithmics 1 module

Cédric Coussinet

Sorting online players scores

You're a team of software engineers working for a video game startup. This startup is focused on fast casual games. The startup's founder is a former chemistry researcher who has turned entrepreneur. She came up with the idea of making short games inspired by molecular combination problems. Winner of the national competition for innovative start-ups, she used the funds to hire you to make a prototype.

After many difficulties, you have succeeded in making this prototype.
But now it's time to work on the community aspect, by promoting real-time competition.

A player must complete ten scores within five minutes. Each score is a real value between 10.0 and 0.0 inclusive. The ten scores, once obtained, are displayed in order of performance and then compared with the scores of the other players to give an overall ranking between the players. The overall ranking is based on the calculation of a final score, which corresponds to the subtraction of the average score by the number of other players' scores falling between the player's maximum and minimum scores, divided by the total number of scores other than the player :

Global_Score =Average_10Scores - nbr_scores_between_min_max /nbr_scores_other_players

For example :

Player	B	A	B	B	B	A	B	A	A	B	B	B	A	A	B	A	B	A	A	A
Score	9.2	9.1	9	8	7.1	7	6.9	6.8	6.7	6.6	5	5.5	5.4	5.3	5	4	3	2	1	0

Global_score of the player A = (9.1 + 7 + 6.8 + 6.7 + 5.4 + 5.3 + 4 + 2 + 1 + 0)/10 – 9/10

The case of identical scores is not considered, as for the former chemist this is highly unlikely given the game.

It all seemed quite straightforward with a random sorting algorithm, but the startup's director wanted an optimized solution. She's hoping the game will go viral, so we'll need to keep up with the load on the server side in order to give the overall ranking as quickly as possible.

The former chemist is aware that there are several ways of sorting.

On the other hand, she has retained her scientific rigor. She would like you to *demonstrate* that you have the best solution for each sort, and that you can *show* this empirically via a Python simulation of the whole chain: sorting local scores, then sorting global scores, then sorting the final ranking.

For intellectual property reasons, the startup's director requires that the sorting algorithms be developed without external libraries.

Finally, to budget for the power to be rented, you'll need to propose a forecast curve linking the calculation time of your sorting algorithms to the number of players (maximum 5 million).

Additional resources for tutors

For sorting ten real numbers, insertion sorting is the best, as you can find out empirically and not by complexity with O notation.

This is due to the fixed complexity cost of the other algorithms compared to insertion sorting.

On the other hand, if you want to simulate the sorting of scores from a very large number of players (several million), you'll need much more efficient sorts. Can you find the right sort?

Resources for dealing with the problem situation

Documents

Module 29 : Notions and enumeration
Module 30 : analysis of two interative sorting methods
Module 31 : Notations for asymptotic analysis

Definition

Complexity analysis
(https://fr.wikipedia.org/wiki/Analyse_de_la_complexit%C3%A9_des_algorithmes)
Analyzing the **complexity of an algorithm** is the formal study of the amount of resources (e.g. time or space) required to run it. It should not be confused with complexity theory, which studies the intrinsic difficulty of problems, and does not focus on a particular algorithm.

Sorting algorithm
(https://fr.wikipedia.org/wiki/Algorithme_de_tri)

In computer science or mathematics, a sorting algorithm is an algorithm that organizes a collection of objects according to a given order .
The objects to be sorted are elements of a set with a total order.

Matplotlib : graphical library in Python https://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science/4740942-maitrisez-les-possibilites-offertes-par-matplotlib

Random numbers in Python :
https://docs.python.org/3/library/random.ht

PBL processing shedule:

Schedule of " Outbound " seance and individual work :

Phases and Steps		Tasks
Phase A « Outbound » seance	**1** 10min	*Organize the team :* ▪ *Divide up the essential functions (see page 6)* ▪ *The helmsman takes note of the stages to be completed and stays on course* ▪ *The timekeeper undertakes to keep an eye on the timing*
	2 10min	**Familiaris yourself with the provided document:** ▪ *Everyone flips through the booklet to familiarise himself with its contents.*
	3 10min	**Understand and clarify the problem: from p. 3 :** ▪ What exactly is the problem we are going to address? ▪ The scribe starts to note down what comes up in the discussion (key words, concepts, ideas, etc.).
	4 30min	**Establish a set of tracks to deal with the problem:** ▪ Draw up a list of relevant questions to be answered ▪ Take stock of what the team knows (and doesn't know) ▪ If necessary, draw up a list of simplifications and restrictions to limit the scope of the problem (if necessary, check with the tutor) ▪ Draw up a list of expected outputs ▪ Consider different ways of moving forward with the treatment ▪ The activator initiates and relaunches the discussion when necessary.
	5 20min	**Identify learning outcomes:** ▪ What do we need to (re-)learn/discover to deal with the problem? ▪ What questions should each of us be able to answer at the end of the "RETURN" session? ▪ What do we need to be able to do?
	6 15min	**Draw up an action plan :** ▪ Determine what information needs to be gathered to confirm or invalidate the leads listed. ▪ Draw up a list of tasks to be completed and deliverables to be prepared by each member before the next meeting, ... ▪ The secretary notes down what is decided and arranges for it to be communicated to the other team members.

Phase B Individual work	**7** de 6 à 10 h	**Individual Work :** ▪ Implement the action plan drawn up in step 6: **everyone carries out the work decided and prepares what they are going to bring to the "RETURN" seance.**

Phase C « RETURN » Seance		(details p. 5)

PBL processing shedule:

Timing seance « Return »

Phases and steps		Task
Phase C « RETURN » seance	**8** 10min	**Organizing the team :** ▪ Who does what (functions)? Should responsibilities be changed? Should new functions be assigned? ▪ What output do we need/choose to produce? → deliverable(s) (if necessary, confirm with the tutor). ▪ Planning: what needs to be done in the next stage and how can we best organise the time available? (+ instructions for the timekeeper)
	9 60min	**Validate learning, solutions and deliverables** ▪ Pool what everyone has studied, prepared and brought along. ▪ Examine answers to questions formulated during the "go" session ▪ Propose answers/solutions to the problem situation ▪ Prepare deliverable(s) ▪ Validate collectively
	25min	**QUIZ**
	25min	**QUIZ Correction**

Functions to facilitate the teamwork...

To ensure that the teamwork runs smoothly and efficiently, a little organization is required... Your tutor had given you cards describing the different functions you need to perform in order to achieve this objective.

The back of each card explains the function defined by the card. Examine the cards and distribute the functions among the members. Each person lays out his or her assigned card(s) in front of him or her, so that each member can see who will be responsible for which function(s).

Among the proposed functions, the "**Active Participant**" function must be assumed by each member!

A few functions to be distributed :

Essential functions :

Helmsman 	▪ You monitor the progress of the work. ▪ You ensure that the team follows the steps imposed or decided upon by the team. ▪ You prevent the team from getting sidetracked and wasting time on dead-end paths.
Activator 	▪ You encourage every team member to make an active contribution to the work, and don't forget the scribe or secretary! ▪ If tasks need to be shared out, you make sure that each member contributes equally.
Time keeper 	▪ You ensure that available time is used efficiently. ▪ You draw attention to the risk of falling behind schedule.
Scribe 	▪ On the shared workspace (e.g.: flip chart), you note important ideas, open questions and patterns that emerge during discussions, but without imposing your own points of view. ▪ You manage the flip chart sheets so that useful information is visible to all team members. ▪ And don't forget to take part in the discussions!
Secretary 	▪ You produce a summary of the key points arising from the discussions: those that need to be retained for the continuation of the work. ▪ You record all the information needed to continue the work: decisions made, deadlines set, upcoming meetings, collective and/or individual work plans, etc. ▪ You circulate your productions and other necessary documents to all team members. ▪ And don't forget to take part in the discussions!
Speech circulator 	▪ You make sure that every team member has a say. ▪ You encourage team members in the background to speak up; you don't forget the scribe or the secretary! ▪ You prevent one or other team member from taking the floor to the detriment of the others.
Spokesperson 	▪ You present the status or results of your team's work in a synthetic and comprehensive way, without showing any preference for your own point of view. ▪ You use all the necessary means for effective communication.
Stitch maker 	▪ What has been done? What remains to be done? What do we know and what don't we know? ▪ You help the scribe to record these elements in the shared workspace.
…	*If necessary, add a function that you find useful or necessary.*