

Thanks for taking the Prodigy Algorithmic Design Challenge!

Below is your multi-part mission. The goal of this challenge is to produce a pseudo-code heuristic to solve each step of the challenge, supported by explanations, assumptions, or working code at your discretion.

Please write your answers to all parts clearly, and send to us as a PDF or URL.

0 - You are given a list of tasks $T = T_1, T_2, \dots, T_n$. Each task is to be done on a computer. Some tasks are easy, other tasks are hard. (Some examples of tasks could be typing, or coding, or answering multiple-choice questions.) You also have a set of users U who have different abilities on these tasks.

1 - You want to estimate the difficulty of each task. Each user is given a random sample of tasks from T to attempt. How can you use the performance of users (on the random task sample they are given) to estimate the difficulty of all the tasks in T ? (You can only use observations on the computational tasks - no biometrics, direct observation, etc. Imagine you can see what is on a user's computer screen but nothing else.)

2 - Now suppose you are given a set of task lists L_1, L_2, \dots, L_n . Each task list contains some tasks in estimated ascending difficulty order as measured by some integer D , where $D = 0$ for an extremely trivial task and $D = 50$ for the hardest task imaginable. However, these given task difficulties are only estimates by experts - in practice each user who tries a task might find that task easier or harder than the expert thought it would be. How would you use the given set of task lists along with the actual performance of users on each task to generate a single combined task list with better estimated difficulties for all tasks?

(Please assume that for any task T_i , the difficulty estimate D_i can vary depending on the specific expert's evaluation of difficulty for T_i - that is to say, experts will likely disagree on how difficult any given task is. You can also assume that the same task may appear on multiple task lists.)

3 - Based on your answer for 2 (i.e. given a single combined task list with estimated difficulties), suppose you wanted to build an adaptive algorithm to order the sequence of tasks each user is given. You want to create a custom list of tasks for each user. This custom list should give tasks of a suitable level of difficulty for each user, and you want the difficulty of tasks each user does to gradually increase over time. How would you approach this?

4 - How would you modify your algorithm from part 3 to estimate a user's frustration level F where $F = 0$ denotes no frustration and $F = 50$ denotes the highest level of frustration a user can safely experience without giving up at each point in time?

Aim for clarity and insight in what you write, and treat this as a fun learning opportunity - the kind of opportunity you'll be helping to build for the world if you join us at Prodigy!