Project Selection

This is a **regular task**. You must submit a PDF, which can be produced using the LaTeX template on Moodle, exported from a word processor, hand-written or any other method.

You have some free time and have decided to improve your coding skills by completing a series of personal projects.

After a brainstorming session, you have come up with a list of n projects. For project i, the minimum knowledge level required to start the project is given by r_i, and the amount of knowledge you will gain by completing the project is given by g_i.

You begin with an initial knowledge level of s. Unfortunately, you only have enough time to complete exactly $k < n$ projects before you will again be inundated with coursework.

We are looking to find the sequence of projects you can choose to maximise your final knowledge level.

(a) First, assume that we can repeat projects as many times as we would like. Design an efficient algorithm to find the sequence of projects that maximises your final knowledge level. Provide time complexity analysis and *brief discussion of correctness*.

(b) Now, assume projects cannot be repeated. Design an algorithm that runs in $O(n \log n)$ time to find the sequence of projects that maximise your final knowledge level.

(c) Using an exchange argument or otherwise, prove that your algorithm is part (b) correct.

Note: For all parts, the values s, r_i, and g_i are all positive, and you are guaranteed that there is a sequence of k projects that you can complete.

Advice.

(a) For correctness, this part only requires a short paragraph informally explaining why you think your algorithm is correct. The full proof will be seen later in (c).

(b) What is done in part (b) compared to part (a) is called relaxing a constraint. This is a useful problem-solving technique that focuses on developing logic through solving a simpler version of the sample problem. How is this useful here?

Expected length:

- For (a), up to half a page.
- For (b), up to half a page.
- For (c), up to half a page.

Solution.

Attribution.