Caffeine

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.

In this task, we will design an algorithm to solve a problem, then apply the idea of inversions to prove the correctness of our algorithm using an exchange argument. Using inversions in exchange arguments is a common pattern: we've already seen it in the *Minimising Job Lateness* and *Tape Storage* lecture problems.

> **Reminder:** in a sequence $S_1, S_2, \ldots, S_n$, an inversion is a pair of indices i and j where $i < j$, but $S_i > S_j$. Informally, S_i and S_j are 'the wrong way around' with respect to sorted order.

Here's the problem:

> A cafe has n drinks on the menu, the ith of which contains c_i milligrams of caffeine per serving. Buzz has n vouchers, the ith of which can be redeemed for v_i servings of any drink. The terms and conditions on the back of the vouchers state that:
>
> - Each voucher must be used entirely on a single drink, e.g. a voucher of value seven could be used for seven cappucinos, but it could not be used for three lattes and four mochas.
> - Each voucher must be used on a different drink, e.g. two vouchers cannot both be used to buy espressos.
>
> For example, suppose the drinks (say cappucino, latte, mocha and espresso) have caffeine contents of $600, 200, 400,$ and 100 milligrams and Buzz's vouchers have values of $6, 3, 8$ and 6.

i	1	2	3	4
name	cappucino	latte	mocha	espresso
c_i	600	200	400	100
v_i	6	3	8	6

> One possibility is to use voucher 1 for six mochas, voucher 2 for three cappucinos, voucher 3 for eight espressos and voucher 4 for six lattes. The total caffeine content would be

$$\begin{aligned} v_1c_3 + v_2c_1 + v_3c_2 + v_4c_4 &= (6 \times 400) + (3 \times 600) + (8 \times 200) + (6 \times 100) \\ &= 6400 \text{ milligrams.} \end{aligned}$$

> Note that this is *not* the optimal allocation!

Your task is to determine how to spend the vouchers to maximise the total amount of caffeine.

(a) In this problem, both the drinks and the vouchers are given in no particular order. To apply a greedy algorithm, we should choose a sensible order to process them in.

We can order the drinks in ascending order, then assign one voucher to each drink in this ordering. With the drinks in this order, any allocation of vouchers is simply a permutation of the voucher amounts $\{v_1, v_2, \dots, v_n\}$. The first voucher in the permutation is used on the weakest drink, and so on.

The optimal permutation is in ascending order of value; we should use the lowest value voucher on the weakest drink, and so on.

Now, we will use an exchange argument to show that an algorithm that assigns vouchers in this way is optimal. Suppose that G is the your ordering of vouchers, and that A is some other alternative ordering i.e. $G \neq A$. We know that at least one inversion in A must be an *adjacent inversion*, where the two vouchers that are 'out of order' are consecutive with respect to our ordering of vouchers. That is, there is some index i where $A_i > A_{i+1}$.

Prove that if we resolve this inversion (by exchanging A_i and A_{i+1}), then the amount of caffeine will be no worse (i.e. the same or better).

Hint: *You may want to take some ideas from the task Greedy Proof. Define your cost function and show that your exchange is valid and optimal.*

(b) Finally, how can we conclude from the previous parts that our permutation G is optimal?

Advice.

(a) Recall that we gave you the correct ordering for this part. First, try to convince yourself that this is correct in the first place:

- Consider which voucher we should use for the most caffeinated drink.

Then what would an inversion do to go against this heuristic?

(b) Use the results of the previous parts and sound logic to complete the proof.

Expected length: Up to one page in total.