AIML 2024-2025 Coursework

September 2, 2024
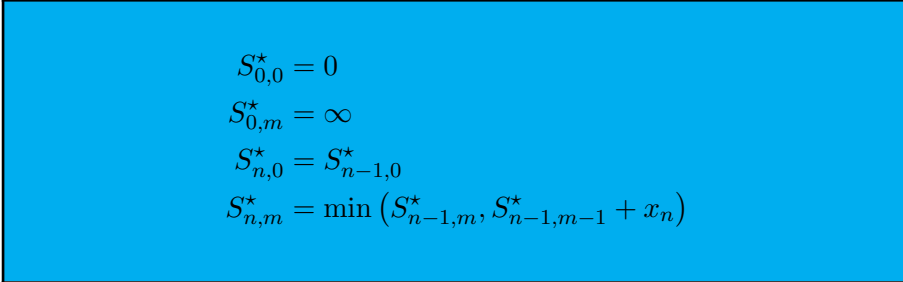
$$S_{0,0}^{\star} = 0$$
$$S_{0,m}^{\star} = \infty$$
$$S_{n,0}^{\star} = S_{n-1,0}^{\star}$$
$$S_{n,m}^{\star} = \min\left(S_{n-1,m}^{\star}, S_{n-1,m-1}^{\star} + x_n\right)$$

Figure 1: Dynamic programming recurrence for computing minimum sum combinations.

Problem The goal of this take-home assignment is to implement, in Python, the recurrence in Figure 1. For an input data list of length N, $[x_1, \ldots, x_N]$, it finds a combination of M elements of smallest sum in $S_{N,M}^{\star}$. As an example, with input $x = [11, 3, -5, 8, 2,]$, then the minimum sum combination at $S_{4,2}^{\star}$ is -3 corresponding to the combination $[-5, 2]$. The input can have duplicate-valued elements.

Instructions The recurrence is to be computed using a function in a single Python file. The interface to the function should be in the **precise format**,

$$S = \mathtt{minsumcomb(x, M)} \tag{1}$$

where `x` is a **Python list** of **numerical values**; `M` is a **positive whole number**; `S` is a **list of lists of type** `argminsum` defined in the course code `argminsum.py`, corresponding to $S^{\star}$. Therefore, when testing, you should expect to import this module.

Submission To prepare the Python code file for submission, it must be named in the format `initials_studentid.py`, for instance if your initials are 'AJD' and your ID is 5716631 then your file should be named `ajd_5716631.py` without any spaces in the file name. Submit the file through the Assignments page on Canvas. The deadline for submissions is **12pm UK time, 5th December 2024**.

Marking Submissions will be marked automatically by calling the function `minsumcomb` inside Python, and checking the results against a model solution. A fully correct solution will receive 20 marks. A solution which has a partially correct interface, or produces only partially correct output values, will lose marks accordingly. It is critical that the interface (1) is correctly implemented. It is critical also that the solution does not have additional code outside the single function (e.g. remove any code used to test your function). There should be no syntax errors which prevent the function from being called. A solution which fails to adhere to these criteria may score 0 marks.