STOR-609 Assessment 1 - Marks and Feedback

March 3, 2025

Name : Malcolm Connoly

1 Quality of Source Code

- Your code (both R and python) is good quality.
- You have supported your code with effective descriptions and comments.

Mark (out of 5) : 5

2 Solution

- For both problems you have provided example code that produces a correct solution.
- Your code is designed so that it was easy for me to change the problem and produce further correct solutions (i.e. your code is both **re-usable** and **re-runnable**)

Mark (out of 5) : 5

3 Understanding Design Principles

- You have provided an implementation for a generic backtracking algorithm which is immediately recognisable as such when compared with the pseudo code provided.
- You have employed several design principles that improve the re-usability of your code, for example, a class template for producing objects which represent different problem types and to encapsulate state.
- You have considered issues regarding global state in your program and attempted to mitigate them.

Notes

- It would be useful for you to consider the computational complexity of your methods in more detail (i.e. how much extra “computation” is required as your problem size increases). This is particularly relevant given that you have used both R an python (the need for deep copies in python may impact on the computational cost).

Mark (out of 5) : 4.5

4 Quality of Written Communication

- You have provided some very high quality supporting material. Importantly, you have discussed your main design choices and reflected on how they impact on **re-susability**.
- You have also considered potential alternatives for your design and contrasted these with the ones you have implemented. This type of reflective practice is of great value (though often overlooked) in software development.

Suggestion

- It would be really good if you also provided pseudo code for each of your **accept**, **reject**, **first**, and **next** methods.

Mark (out of 5) : 4.5

5 Overall Comments and Marks

- Very high quality work. I am impressed by the amount consideration you have given to alternative ways of designing and implementing your code.
- In addition, you have demonstrated your ability to code using two different programming systems and understood dome of the differences in the idioms they use, for example, having to make deep copies of mutable data structures in python when using recursion that "bactracks". Given your interest in other languages you might like to experiment with Julia or, if you really want to push the boat out, F sharp. Both these languages support programming paradigms that are not available in R or python, and F sharp has full support for tail end recursion. However, I do realise you have to be pragmatic given your busy workload !!

Notes

- The recursion limit is not related to the "power" of the computer. It is a consequence of the design of the R programming system. This has some major implications in the sense that many useful algorithms are best expressed recursively. Of course, any recursive algorithm can be expressed using iteration (and vice versa).
- Do you remember looking at bitwise operations in the **Introductory Python Course** ? How might they be useful in the context of this assignment ?

Overall Mark (out of 20) : 19