Demcon festival challenge

# Introduction

Like most scheduling problems, I expect also this one to be [NP-complete](https://en.wikipedia.org/wiki/NP-completeness). So it is possible to solve by use brute force but that would require working out all possibilities.

An option is to create a deterministic algorithm that solves the problem but might not have the best result. Now by introducing some stochastic process on the inputs, we can create a multitude of solutions and select the best one. This will still not be the perfect solution but in most cases it will be good enough.

# Datastructures

We can recognize the following data / objects:

* Show = name + starttime + endtime
* Stage = sorted list of shows (0..n)
* Schedule = list of stages (0..m)

These are used for the classes.

# Deterministic Algorithm

The deterministic algorithm I have chosen, works as follows:

1. Give each show its own stage.
2. Try to combine stages where the shows don’t conflict.

The latter is done by comparing a stage with the others and merge when possible. This is done by starting with the last stage and checking with all previous stages. Then we take the second last stage and do the same, all up to the first stage. In fact, we could also start with the first stage, doesn’t make a difference.

# Stochastic Process

The randomization is done by changing the order of the stages after assigning each show its own stage.

For this the [Fisher-Yates](https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm) algorithm is used.

# Selecting the best schedule

Currently, the selection is simply based on the amount of stages needed. However, more factors could be used.