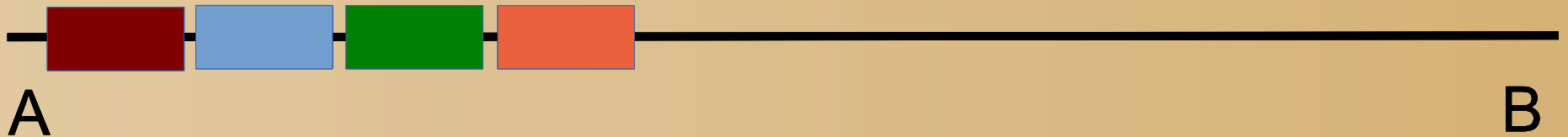
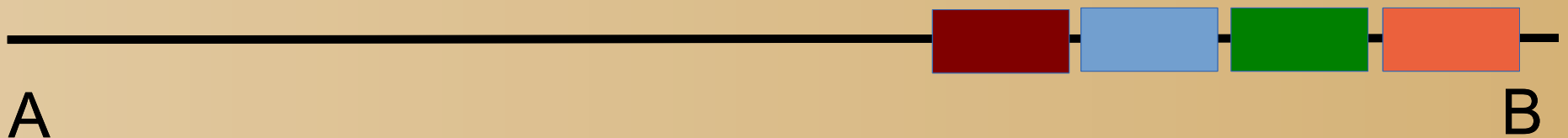


How NXFFS Works

Imagine a train that starts at City A



And travels to City B



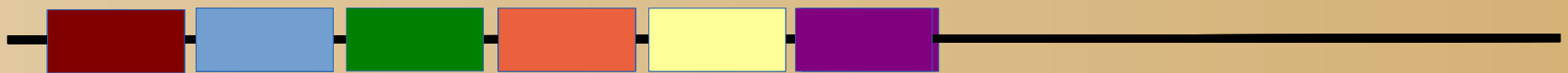
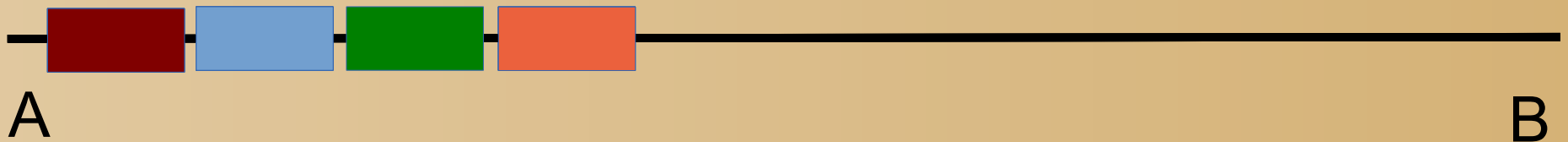
When the train reaches City B. It stops and is pulled back to City A



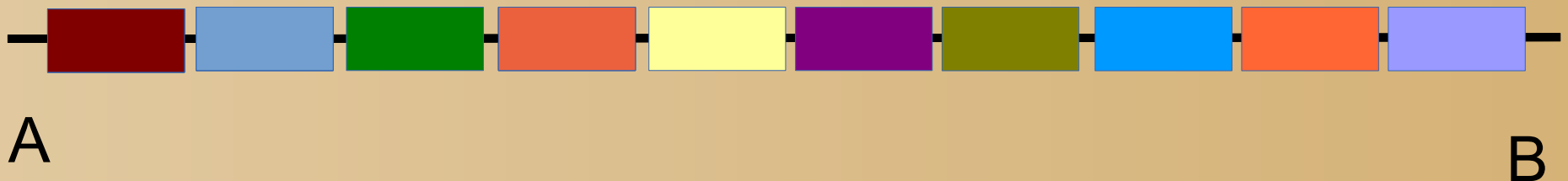
In this analogy, each car in the train is a file. City A is the start of FLASH and City B is the end of FLASH.

Now NXFFS Works

Let's improve the analogy. Imagine that the train does not actually move. Imagine instead that new cars are added to the front of the train



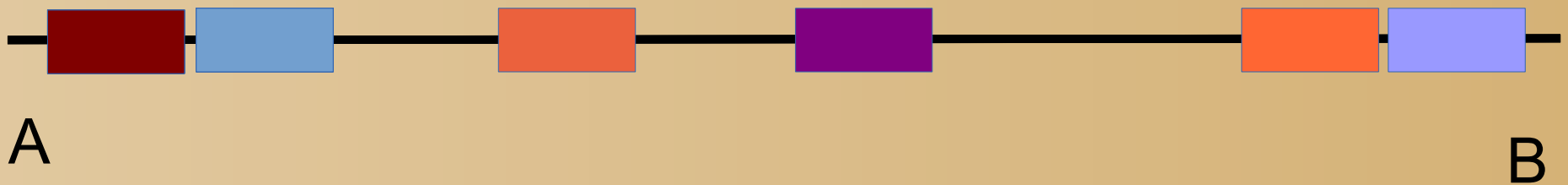
Eventually, enough cars are added to the front of the train, that it arrives at City B (although it hasn't actually moved).



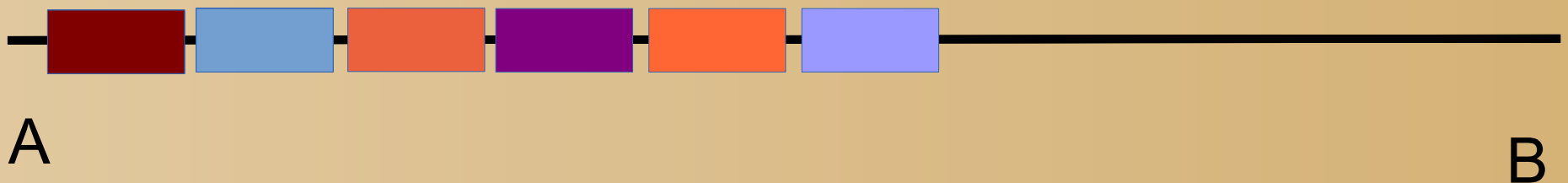
At this point we have train cars (files) stretching all of the way from City A (the beginning of FLASH) to City B (the end of the FLASH). We cannot add even a single new car to the train (because the FLASH is full).

Now NXFFS Works

Now suppose that cars (files) are not just added to the train, but that cars are also removed from the train. This corresponds to files being deleted from FLASH. So when the final car reaches City B, there are empty spaces where the cars have been removed:



Now when the train is pulled back to City A, then spaces are closed.



This is the periodic re-organization that NXFFS needs to do in order to free up additional space for more files to be added to the FLASH.