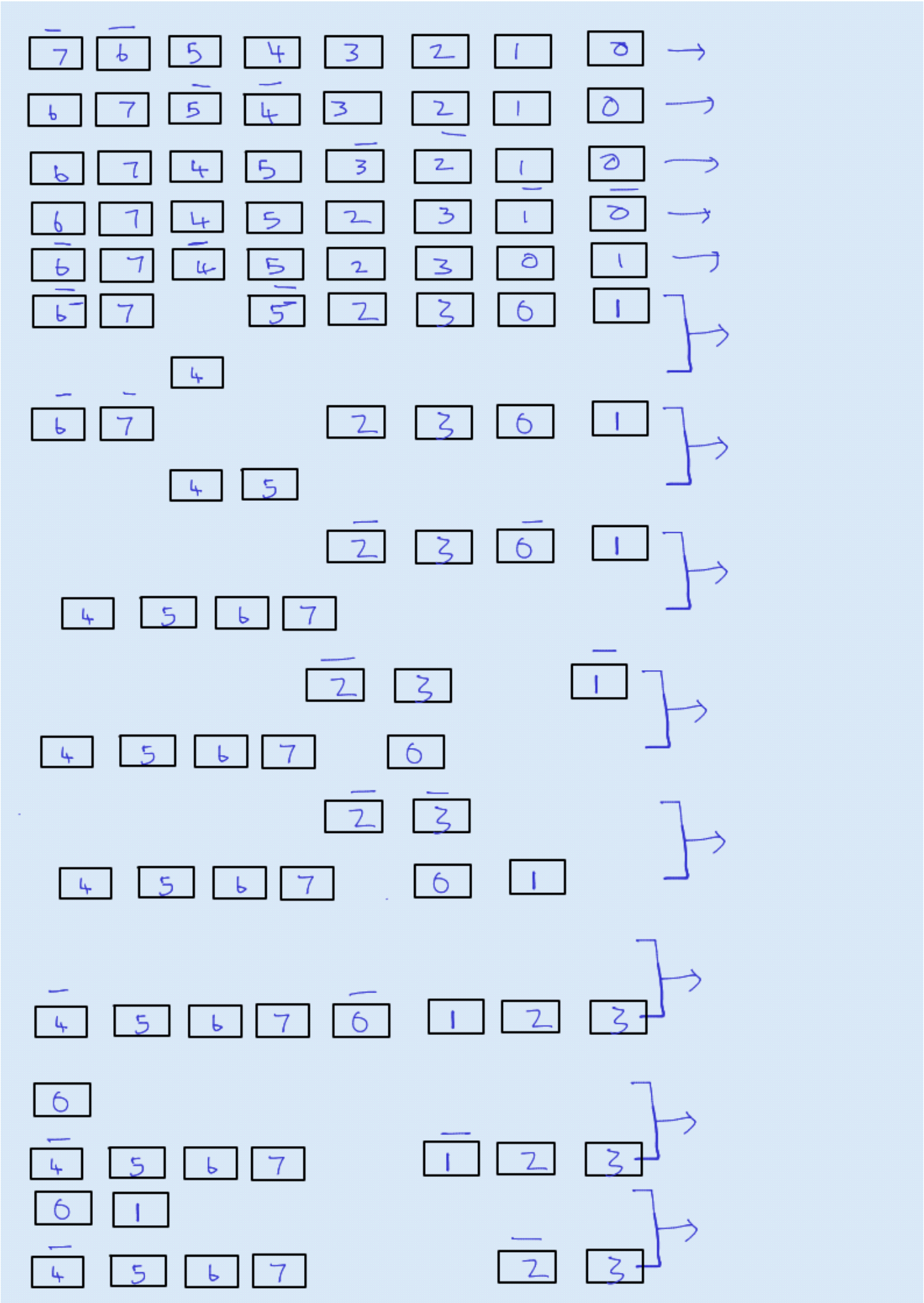
Dialogues for Episode 1

01 October 2020 20:14

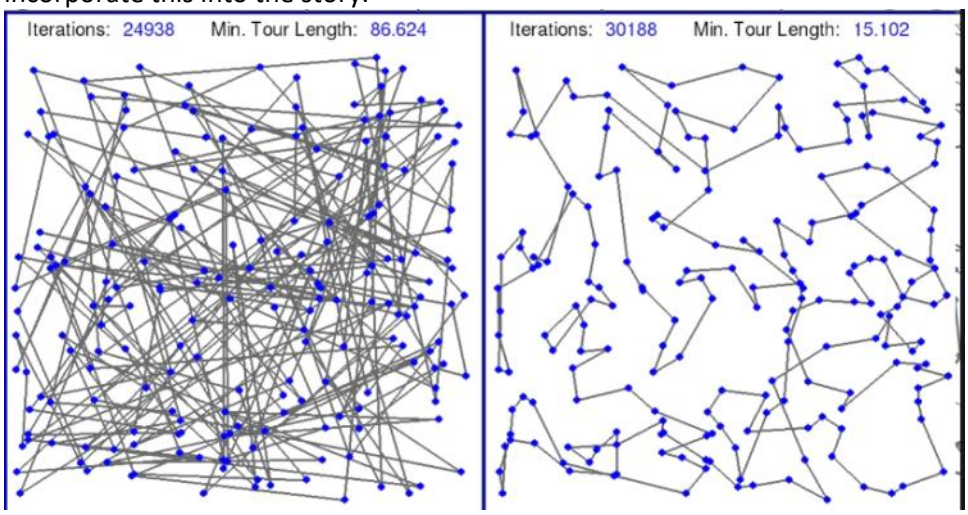
Re-read the whole scene script and made small changes grammatically.Made a visualization of merge sort for the comic.



Dialogues for Episode 2

10 November 2020 18:15

Using the algorithm proposed by the Algo team("Travelling Salesman problem"), I wrote dialogues to incorporate this into the story.



Nodes are the gang hideouts where Jekyll will get a part of his antidote. Jekyll's goal here is to minimise the time (or distance as velocity will be fixed) to collect all the parts of his antidote. As the algorithm is too complex to imagine and calculate mentally, he takes help of his laptop to compute the best path.

Dialogues for Episode 3

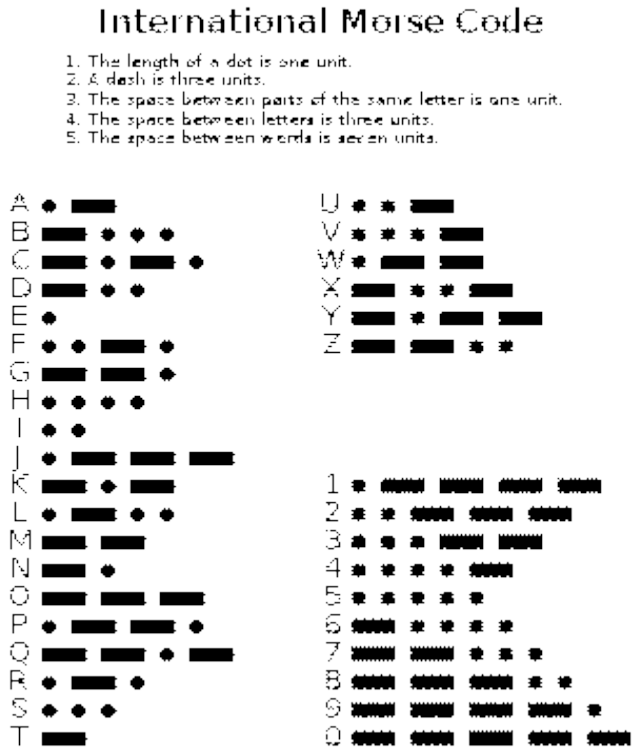
21 November 2020 14:08

Researched a little bit about the nitty-gritties of Knapsack to find a suitable situation to incorporate the algorithm with the story. After a discussion with Arjun and Gokul, we decided to simulate knapsack in file transferring from servers.

Dialogues for Episode 4

21 November 2020 14:11

As this was planned to be the finale, the algorithm chosen and the situation were tweaked in a way to make the solution more subtle than usual. I had to research a bit about communication codes to encrypt the audio signal in. The obvious choice was Morse code.



As this was way too mainstream, I thought of switching to more subtler ones like the tap codes which are usually used as a communication medium between prisoners. There are many types, but I decided to go with the most common 5x5 tap code. This is based on a Polybius Square using a 5x5 grid. One fact which is exploited in the story is that 5x5 tap code doesn't have an unique code for letter 'k', it has the same value as 'c'. Only people well versed in this can notice this subtle difference.

	1	2	3	4	5
1	A	B	C/K	D	E
2	F	G	H	I	J
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Letter 'X' is used to break up sentences but not used in the story. In the story, the word "Jekyll" is encrypted in tap code and the resultant read would be "Jecyll". Searched for KMP writeups for the appendix which was easy to understand and clear to understand.

https://www.inf.hs-flensburg.de/lang/algorithmen/pattern/kmpen.htm
https://www.topcoder.com/community/competitive-programming/tutorials/introduction-to-string-searching-algorithms/