CS-293 Course Project

- Steganography
- Indian Railway Reservation System Simulation

•Gautam Sumu (100050058)
•Chandan Kumar (100050059)

•Instructor- Prof. Varsha Apte
•TA- Mayur

Steganography

- You can hide a secret message (an image or a text) into another image and retrieve it back, when required (without letting others know that there is something fishy, of course :P)
- There is an extra feature of password as well, so that you can protect your secret messages further.

Algorithm

- For hiding replace the lower bits of the cover image with the higher bits of the secret image.
- While extraction , we extract the lower four bits of the encrypted image. This gives us the secret image.
- For example:

  Sorce pixel: 10100010

  Secret pixel: 11001010

  Result pixel: 10100000 (from source) + 1100 (from secret) = 10101100

- For hiding text in image, we hide ASCII value of each character across the RGB values of a pixel.
- For example, say ASCII value is 1011001 (89).

  Then lower 3 bits of R becomes 101,

  lower 2 bits of G becomes 10,

  and lower 2 bits of B becomes 01.
- Hence we can store a character (max. 7 bits) in a pixel.
- While extraction, we calculate the ASCII values accordingly.
- Using the same method, password is also stored and checked.

Libraries used and class design

- We have used ‘Cimg’ library and ‘Image-Magick’ package for reading, editing and writing the image pixel values from and to files.
- We have the code in three files- ‘stego.cpp’, ‘stego.h’ and ‘main.cpp’.
- We have created a class ‘Servant’. This class basically contains the functions used for encrypting, decrypting images/text. An object of this class is used to do all the tasks.

Work division

- Gautam: Reading, writing images and text files.
- Chandan: Functions for encryption, decryption and password implementation.
- Other parts were done by both.

Indian Railway Reservation System Simulation

Railways are the backbone of our country. Its proper utilization is extremely important for our progress.

Our project has been created with this goal in mind. It takes various inputs

- (no. of trains, train nos. (different for each train),
- departure day and time for each train,

- No. of AC and SL berths and their fare respectively,
- Customer preference for each train (on a scale of 1 to 10).
- Running cost for each train.
- Trace (0 or 1) for debugging purpose.

Special features of our project:

- It takes into account, the preference of customers for each train.
- Reservation and cancellation rates for seats in a train increase as departure date comes closer.
- Reservation for a train is closed before two hours of departure.

And after execution, we get various values like:

- Financial data for each train (profit or loss).
- No. of berths filled and no. of customers who remained in waiting list.
- Three classes of customers are generated: first class of customer always looks for AC tickets, the second class customers somewhat have flexible choice. They prefer AC berths and the third class always books SL berths (can't afford AC).

Algorithm

- Customers are generated randomly generated per train and their type is decided.
- According to their type they book the berth on train.

Class design

- There are four classes:
- Class Train :- It contains all the data related to train. It also operates the booking and cancellation function.
- Class Event :- It stores all the information about a happening of event.
- Class Eventlist :- It is a priority queue which returns the latest event to take place.
- Class Simulator :- Its main purpose is to simulate the reservation simulation.

Work division

- Chandan: version 1 with one train, one day, and random customers. Extending this program for multiple days, multiple trains, stopping reservation before 2 hours of departure.
- Gautam: extending the program to add preference features to trains, adding the feature of cancellation of tickets. Also the feature of increase in frequency of reservations and cancellations as the departure time comes closer.

Thank You ☺