VISTARA AIRBUS QUEUE MANAGEMENT

ALGORITHM AND SOURCE CODE

Author: Sagnik Mitra
Department of Computer Science & Business Systems
Sister Nivedita University [2018 - Present]

About this Project:

This was basically a challenge in the TCS Hackathon but due to some unavoidable circumstances, it could not be finished within the due time. So, the part of the problem that I solved is kept here as a self-explanatory tool.

As I see in the Airports, there is a lot of rush in each and every Checking-in & Boarding scenario. So, through this project from Hackathon, I am trying to make an optimized way to maintain the rush.

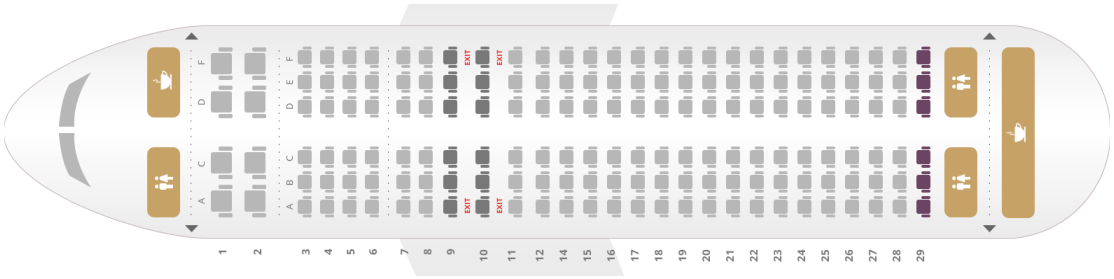
So, I built a very affordable and relatively simple Queue Management System for the same. At first, it is clear that there must be different processes to maintain different queues like Boarding Queue, Checking Queue & On-Board Queue. I was able to innovate the solution for the Boarding Queue, so I reflected it here.

Boarding Queue Management:

As I know, there are three types of Vistara Airbus. Airbus A320, Airbus Economy A320 & Boeing 737-800NG. Here, I primarily took A320 for our case.

Point to Notice → The Business Class (1A-2F: 6 Passengers) followed by the Premium Economy Class (3A-6F:24 Passengers) will be Boarded the First Row by Row to make the Pathway clear for a large number of Economy Class Passengers.

This is a picture of the seating arrangement from the Vistara Website
https://www.airvistara.com/ae/contents/ae/s3fs-public/Airbus-320.png



Now the next whole thing is for the Economy Class. This Seat Arrangement is taken from the VISTARA AIRBUS A320. If you are dealing with Airbus All Economy, simply replace n>=7 by n>=1.

In this part, considering there are 22 rows for the economy class, I set to the gate to be the entrance for separated parts of seats, like Gate 1 is for 7A-14F & Gate 2 is for 15A - 29F passengers. So, here I created a Q-Card / Q-Code to make boarding an optimized process by managing the queue to the flight from the gate.

Here is the pseudocode for generating the last part of the Q-Code and the full format is explained at the end of the document.

```
whether n >= 7
        If yes
           whether n <=14
                If yes
                 whether n<14
                        If yes
                            //Being treated as Unlucky 13, after 12A-F, 14A-F is there in Vistara
                            nA → [27-2*n]W ; nB → [27-2*n]M ; nC → [27-2*n]S;
                            nF → [28-2*n]W ; nE → [28-2*n]M ; nD → [28-2*n]S;
                        If no
                            14A → 1W ; 14B → 1M ; 14C → 1S ;
                            14F → 2W ; 14E → 2M ; 14D → 2S ;
                If no
                        nA → [2*n-29]W ; nB → [2*n-29]M ; nC → [2*n-29]S ;
                        nF → [2*n-28]W ; nE → [2*n-28]M ; nD → [2*n-28]S ;
```

Methods Used:

1. ***Negotiation & Decision Technique for detecting the proper Seat Number***
2. ***Odd Number-Even Number Combinatorics for getting the Q-Number as Ill as the Q-Code as mentioned below.***

Q CODE Format -> Q / 10F / G1 / 8W

F → Whether it is a Window (W) or Middle (M) or Side Seat(S).

Q → Stands for Queue Code for Vistara Air

10 → 10th Row from the front.

G1 → If your Seat Row is >14, then you have to take the 2nd Gate from the Pilot's POV And if your Seat Row is <= 14, then take the First Gate from the Pilot's POV.

8W → FolloId from the previously mentioned process.

Process:

When the Flight Officials announce the time for boarding, there will be two lines simultaneously, one for the Gate 1 & One for the Gate 2. According to their Q Code,

The passengers of the first line will be in the following sequence according to their Q Card. ->

G1/1W, G1/2W, G1/3W, G1/4W, ... , G1/11W, G1/12W, G1/13W, G1/14W
(14A, 14F, 12A, 12F, ... , 8A, 8F, 7A, 7F)

Then, G1/1M, G1/2M, G1/3M, G1/4M, ... , G1/11M, G1/12M, G1/13M, G1/14M
(14B, 14E, 12B, 12E, ... , 8B, 8E, 7B, 7E)

Then, G1/1S, G1/2S, G1/3S, G1/4S, ... , G1/11S, G1/12S, G1/13S, G1/14S
(14C, 14D, 12C, 12D, ... , 8C, 8D, 7C, 7D)

And simultaneously, the passengers of Gate 2 in 2nd Line will be in the following sequence →

G2/1W, G2/2W, G2/3W, G2/4W, ... , G2/11W, G2/12W, G2/13W, G2/14W
(15A, 15F, 16A, 16F, ... , 28A, 28F, 29A, 29F)

Then, G2/1M, G2/2M, G2/3M, G2/4M, ... , G2/11M, G2/12M, G2/13M, G2/14M
(15B, 15E, 16B, 16E, ... , 28B, 28E, 29B, 29E)

Then, G2/1S, G2/2S, G2/3S, G2/4S, ... , G2/11S, G2/12S, G2/13S, G2/14S
(16C, 16D, 16C, 16D, ... , 28C, 28D, 29C, 29D)

Why can QCode be efficient enough to handle boarding Queue?

Answer:

- The format is truly significant with respect to both the boarding and seat arrangement of the Airbus.
- The cost of production of **QCode** will be as low as luggage coupons in Shopping Malls.
- The passengers themselves will be aware of when to board.
- Fourthly, the serialization of boarding into the plane and dividing the passengers for the two gates will help in reducing the rush.

Proposed Look (2.5" x 2.5")

The QCode(s) will be collected after the plane takes off.

So, keeping in mind the vital points like Passenger Friendly, Staff Friendly, Cost Effectiveness, Ease of Implementation, Methodically Sound, I think this will work for managing the whole queue in a befitted manner.

References:

1. An Introduction to Theory of Numbers - G.H. Hardy [Wiley]
2. https://www.geeksforgeeks.org/fundamentals-of-algorithms/
3. https://www.airvistara.com/trip/seatmap/airbus-A320
4. https://www.airlinequality.com/airline-reviews/vistara/
5. https://www.airvistara.com/ae/contents/ae/s3fs-public/Airbus-320.png
6. https://www.underconsideration.com/brandnew/archives/new_logo_and_livery_for_vistara_by_brand_union_raykeshavan.php

Acknowledgment:

I am thankful to Sneharup Mukherjee & Spandan Pal for helping me in all aspects of this project.

Source Code

AIRBUS A320 IMPLEMENTATION

```
//VISTARA BOARDING QUEUE Q-CODE FOR ECONOMY CLASS AIRBUS A320
//A C++ IMPLEMENTATION BY SAGNIK MITRA
#include <iostream>
using namespace std;
#define start_row_for_A320 7
int main(void)
{
   int s_row, i, j;
   char s_position;
   cout << "-----------------------------------------------------------------------------------------------------" << endl
        << "VISTARA AIRBUS A320 BOARDING Q-CODE GENERATION:" << endl
        << "-----------------------------------------------------------------------------------------------------" << endl;
   //TAKING SEAT NUMBER INPUT FROM USER OF ECONOMY CLASS
   cout << "Instructions for giving an input:" << endl
        << "-----------------------------------------------------------------------------------------------------" << endl
        << "For example, if your Seat Number is 23F, then" << endl
        << "On first query (i.e. Enter your Seat Row: ), you will give input 23 as per the demo seat number and"
<< endl
        << "On the second query (i.e. Enter your Seat Position: ), you will give input F as per the demo seat
number" << endl
        << "Otherwise, you will get an Invalid Input Prompt." << endl
        << "The Seat Position Input is not case sensitive." << endl
        << "-----------------------------------------------------------------------------------------------------" << endl
        << "Enter your Seat Row [The Numeric Digit(s) on the left of your seat no.]: ";
   cin >> s_row;

   //INVALID SEAT ROW DETECTION
   if (s_row >= 1 && s_row <= 6)
   {
      cout << "\nYou're a Non-Economy Class Passenger, Q-Code Generation Process is for Economy Class
Only." << endl
           << "Economy Class Seats at VISTARA AIRBUS A320 are ranged from 7A to 12F & 14A to 29F. Thank
You.\n";
      exit(1);
   }

   else if (s_row < 1 || s_row > 29 || s_row == 13)
   {
      cout << "Invalid Seat Row. Economy Class Seats at VISTARA AIRBUS A320 range from 7A to 12F & 14A
to 29F. Thank You.\n";
      exit(1);
   }
```

```
   cout << "Enter your Seat Position [The Alpha Character(s) on the right of your seat no.]: ";
   cin >> s_position;

   //INVALID SEAT POSITION DETERMINATION
   if (s_position != 'a' && s_position != 'b' && s_position != 'c' && s_position != 'd' && s_position != 'e' &&
s_position != 'f' && s_position != 'A' && s_position != 'B' && s_position != 'C' && s_position != 'D' &&
s_position != 'E' && s_position != 'F')
   {
      cout << "Invalid Seat Position. Economy Class Seats at VISTARA AIRBUS A320 range from 7A to 12F &
14A to 29F. Thank You.\n"
           << endl;
      exit(1);
   }
   //UPPERCASE UPDATION
   if (s_position == 'a')
      s_position = 'A';
   if (s_position == 'b')
      s_position = 'B';
   if (s_position == 'c')
      s_position = 'C';
   if (s_position == 'd')
      s_position = 'D';
   if (s_position == 'e')
      s_position = 'E';
   if (s_position == 'e')
      s_position = 'E';
   cout << "-----------------------------------------------------------------------------------------------------" <<
endl;
   cout << "Your Q-Code is: Q/" << s_row << s_position << "/";

   //GATE DETERMINATION
   if (s_row >= start_row_for_A320 && s_row <= 14)
   {
      cout << "G1";
   }
   if (s_row >= 15 && s_row <= 29)
   {
      cout << "G2";
   }
   cout << "/";

   //QCODE NUMERIC CHARACTER DETERMINATION
   if (s_row >= start_row_for_A320)
   {
      if (s_row <= 14)
      {
         if (s_row < 14)
         {
            if (s_position == 'A' || s_position == 'B' || s_position == 'C')
            {
```

```
            int temp1 = (27 - (2 * s_row));
            cout << temp1;
        }
        if (s_position == 'D' || s_position == 'E' || s_position == 'F')
        {
            int temp2 = (28 - (2 * s_row));
            cout << temp2;
        }
    }
    else if (s_row = 14) //WORKING FINE
    {
        if (s_position == 'A' || s_position == 'B' || s_position == 'C')
        {
            int temp3 = (28 - (2 * s_row) + 1);
            cout << temp3;
        }
        if (s_position == 'D' || s_position == 'E' || s_position == 'F')
        {
            int temp4 = (28 - (2 * s_row) + 2);
            cout << temp4;
        }
    }
  }
  else if (s_row >= 15 && s_row <= 29)
  {
    if (s_position == 'A' || s_position == 'B' || s_position == 'C')
    {
        int temp5 = ((2 * s_row) - 29);
        cout << temp5;
    }
    if (s_position == 'D' || s_position == 'E' || s_position == 'F')
    {
        int temp6 = ((2 * s_row) - 28);
        cout << temp6;
    }
  }
}

//Q-CODE ALPHA CHARACTER DETERMINATION
if (s_position == 'A' || s_position == 'a' || s_position == 'F' || s_position == 'f')
  cout << "W";
if (s_position == 'B' || s_position == 'b' || s_position == 'E' || s_position == 'e')
  cout << "M";
if (s_position == 'C' || s_position == 'c' || s_position == 'D' || s_position == 'd')
  cout << "S";
cout << endl
   << "-----------------------------------------------------------------------------------------------------" << endl
   << "Please maintain the queue to the Boarding Gate as per your generated Q-Code." << endl
   << "Thank You. Have a safe journey." << endl;
cout << "-----------------------------------------------------------------------------------------------------" << endl
```

```
      << "A C++ IMPLEMENTATION BY SAGNIK MITRA, B.TECH, CSBS, SNU." << endl;
   cout << "-------------------------------------------------------------------------------------------------------" <<
endl;
}
```

AIRBUS ALL ECONOMY IMPLEMENTATION

```
//VISTARA BOARDING QUEUE Q-CODE FOR ECONOMY CLASS AIRBUS ALL ECONOMY
#include <iostream>
using namespace std;
#define start_row_for_all_economy 1
int main(void)
{
   int s_row, i, j;
   char s_position;
   cout << "-------------------------------------------------------------------------------------------------------" << endl
      << "VISTARA AIRBUS ALL ECONOMY BOARDING Q-CODE GENERATION:" << endl
      << "-------------------------------------------------------------------------------------------------------" << endl;
   //TAKING SEAT NUMBER INPUT FROM USER OF ECONOMY CLASS
   cout << "Instructions for giving an input:" << endl
      << "-------------------------------------------------------------------------------------------------------" << endl
      << "For example, if your Seat Number is 23F, then" << endl
      << "On first query (i.e. Enter your Seat Row: ), you will give input 23 as per the demo seat number and"
<< endl
      << "On the second query (i.e. Enter your Seat Position: ), you will give input F as per the demo seat
number" << endl
      << "Otherwise, you will get an Invalid Input Prompt." << endl
      << "The Seat Position Input is not case sensitive." << endl
      << "-------------------------------------------------------------------------------------------------------" << endl
      << "Enter your Seat Row [The Numeric Digit(s) on the left of your seat no.]: ";
   cin >> s_row;

   //INVALID SEAT ROW DETECTION
   if (s_row < 1 || s_row > 29 || s_row == 13)
   {
      cout << "Invalid Seat Row. Economy Class Seats at VISTARA AIRBUS ALL ECONOMY are ranging from
1A to 12F & 14A to 29F. Thank You.\n";
      exit(1);
   }
   cout << "Enter your Seat Position [The Alpha Character(s) on the right of your seat no.]: ";
   cin >> s_position;

   //INVALID SEAT POSITION DETERMINATION
   if (s_position != 'a' && s_position != 'b' && s_position != 'c' && s_position != 'd' && s_position != 'e' &&
s_position != 'f' && s_position != 'A' && s_position != 'B' && s_position != 'C' && s_position != 'D' &&
s_position != 'E' && s_position != 'F')
   {
```

```
      cout << "Invalid Seat Position. Economy Class Seats at VISTARA AIRBUS ALL ECONOMY are ranging
from 1A to 12F & 14A to 29F. Thank You.\n"
           << endl;
      exit(1);
   }

   //UPPERCASE UPDATION
   if (s_position == 'a')
      s_position = 'A';
   if (s_position == 'b')
      s_position = 'B';
   if (s_position == 'c')
      s_position = 'C';
   if (s_position == 'd')
      s_position = 'D';
   if (s_position == 'e')
      s_position = 'E';
   if (s_position == 'e')
      s_position = 'E';

   cout << "-----------------------------------------------------------------------------------------------------" <<
endl;
   cout << "Your Q-Code is: Q/" << s_row << s_position << "/";

   //GATE DETERMINATION
   if (s_row >= start_row_for_all_economy && s_row <= 14)
   {
      cout << "G1";
   }
   if (s_row >= 15 && s_row <= 29)
   {
      cout << "G2";
   }
   cout << "/";

   //QCODE NUMERIC CHARACTER DETERMINATION
   if (s_row >= start_row_for_all_economy)
   {
      if (s_row <= 14)
      {
         if (s_row < 14)
         {
            if (s_position == 'A' || s_position == 'B' || s_position == 'C')
            {
               int temp1 = (27 - (2 * s_row));
               cout << temp1;
            }
            if (s_position == 'D' || s_position == 'E' || s_position == 'F')
            {
               int temp2 = (28 - (2 * s_row));
```

```
            cout << temp2;
          }
        }
        else if (s_row = 14) //WORKING FINE
        {
          if (s_position == 'A' || s_position == 'B' || s_position == 'C')
          {
            int temp3 = (28 - (2 * s_row) + 1);
            cout << temp3;
          }
          if (s_position == 'D' || s_position == 'E' || s_position == 'F')
          {
            int temp4 = (28 - (2 * s_row) + 2);
            cout << temp4;
          }
        }
      }
      else if (s_row >= 15 && s_row <= 29)
      {
        if (s_position == 'A' || s_position == 'B' || s_position == 'C')
        {
          int temp5 = ((2 * s_row) - 29);
          cout << temp5;
        }
        if (s_position == 'D' || s_position == 'E' || s_position == 'F')
        {
          int temp6 = ((2 * s_row) - 28);
          cout << temp6;
        }
      }
    }

    //Q-CODE ALPHA CHARACTER DETERMINATION
    if (s_position == 'A' || s_position == 'a' || s_position == 'F' || s_position == 'f')
      cout << "W";
    if (s_position == 'B' || s_position == 'b' || s_position == 'E' || s_position == 'e')
      cout << "M";
    if (s_position == 'C' || s_position == 'c' || s_position == 'D' || s_position == 'd')
      cout << "S";

    cout << endl
       << "-----------------------------------------------------------------------------------------------------" << endl
       << "Please maintain the queue to the Boarding Gate as per your generated Q-Code." << endl
       << "Thank You. Have a safe journey." << endl;
    cout << "-----------------------------------------------------------------------------------------------------" << endl
       << "A C++ IMPLEMENTATION BY SAGNIK MITRA, B.TECH, CSBS, SNU." << endl;
    cout << "-----------------------------------------------------------------------------------------------------" <<
endl;
}
```