Final Project Report

Algorithm Design - FUV Course Scheduling

By Thuan Than, Hong Ha, Quan Bui, Tu Ly, and Tan Nguyen

About the Problem

At the moment, Fulbright is still using a manual process to arrange the course schedule for each semester. With the growing number of courses per semester and many new constraints, this task has become more challenging to be done by hand.

Thus, the proposed problem focuses on building an algorithm that can automatically schedule courses for a semester at Fulbright.

For the scope of the algorithm, we will use 3 inputs:

- A list of courses at FUV, including attributes: the name of the course, the level of the course, and the course instructor(s)
- A list of available rooms at FUV that can be used for classes, including attributes: the room name and its maximum capacity
- A list of possible time slots for classes

The output will be either: A schedule if it's possible, or A "No" if it's impossible and some suggestions for reducing constraints

The main idea of the algorithm

We started approaching the problem by thinking about the simplest way to solve it. We arrived at the naive algorithm, which works as described in appendix 1.

Even though the naive algorithm works, it has a time complexity $C_{mp}^{n} * n! * n^2$, which is very slow. We changed to a genetic algorithm approach. The new approach works well and can produce a schedule for small-to-medium test cases (around 25-30 courses), however, it breaks when the test cases become larger.

We had to shift gears again, and now coming up with a new approach called "Heuristic Strategy". It works like this:

First, we divide all possible timeslots into 3 groups: Group 1 contains all timeslots on Monday and Wednesday, Group 2 contains all timeslots on Tuesday and Thursday, and Group 3 contains all timeslots on Friday. The rationale behind this is, typically, classes at Fulbright are either divided up into 2 sessions per week with 1.5h per session, or it can be in one 3-hour session on a single day (usually Friday).

Second, we now combine the list of available classrooms and groups of timeslots to become different combinations. This is the solution space.

Combinations for Mon-Wed slots	Combinations for Tue-Thu slots	Combinations for Fri slots

Next, for every faculty member, we count how many courses they teach in that semester. We detected a pattern in which there are 3 scenarios: faculty with 1 course, faculty with 2 courses, and faculty with 3 courses.

We prioritize faculty with 3 courses to arrange their course schedule first, for they are easy to have a conflicting schedule. Each of their courses will be assigned to a type of combination (as seen above).

Next, for faculty with 2 courses, we arrange them in combination types 1 and 2 to achieve a more symmetrical schedule.

Lastly, for faculty with only 1 course, we assign them to the remaining combinations.

That's the big picture. It's worth noting that, during the arranging process, there are a few rules we create to achieve a more realistic schedule. These rules are listed in appendix 2.

Code instruction

Step 1: Download java (You can refer to appendix 3 for the download instruction)

Step 2: Download the folder code on Canvas. In the folder, you will see both the folder for codes and the folders for test cases.

Step 3: Open Visual studio code —> Select "File" —> Select "Open folder" —> Select the newly downloaded folder "Fulbright-Course-Scheduling" —> "Select folder" to open all the sub-folders in there.

Step 4: Click on file "main.java"

Step 5: To change between the test cases, please fix these 3 lines of code below to the address of the test case (in the folder) you want, but don't change the file name

Private static final String COURSE_OFFERING_PATH = [use the file name of "Course.csv"]

Private static final String ROOM_PATH = [use the file name of "Room.csv"]

Private static final String CLASS_TIME_PATH = [use the file name of "Time.csv"]

Please be careful to use the same test cases for all 3 lines.

Step 6: Select "Run and Debug" to run the program. The program will turn back a list of courses already assigned the time and room. The order of the result is displayed below:

Example of a result: ARTS101 Introduction to Visual Studies [Richard Streitmatter-Tran] 50 false [Mon, Thu] 15:00 16:30 Ting Foundation 60

Interpretation: Course Code Course name [Faculty name] Course capacity Status of online (False==No) [Date assigned] Time assigned Room assigned Room capacity

Step 7 (optional): Visualize the schedule - follow the instruction in appendix 4

Appendix 1: Naive Algorithm

How our naive algorithm works:

- First, generate a set A including all possible combinations of a time slot and a classroom.
- Generate a set B including all subsets of A that have exactly n items.
- Generate a set C including all permutations of given courses.
- Generate a set D including all possible schedules by matching 1 – 1 each element in B and C.
- Loop into each schedule in D, and then loop into each pair of courses in that schedule to find a schedule that does not violate any constraints.

(Time complexity: $C_{mp}^{n} * n! * n^2$, where n is the number of courses, m is the number of timeslots, and p is the number of rooms)

Appendix 2: A few selection rules

1. Space optimization: Courses with small capacity can be put in both small rooms and big rooms, while courses with large capacity can only be put in big rooms. Thus, we prioritize matching small courses with small rooms to leave big rooms for big courses. The size of a course is taken from the attribute "Capacity", not from the course level to ensure realistic input.
2. Special rooms assignment: There are certain courses that need to use special facilities. Thus, we follow 2 rules here:
    i. For CORE105 and engineering courses (course codes start with ENG) - they are prioritized for Maker Space
    ii. For CORE104 and science courses (course codes start with IS) - they are prioritized for Science Lab
3. Online courses come last: Online courses do not use space facilities, so we assign time for them last.

As we already have a schedule of room assignments for courses, we can see a possible timeslot for the faculty weekly meeting is from 11:00 - 13:00 every Friday. There are no classes happening during this time.

Appendix 3: Java installation instruction

Windows & macOS

If you are using Windows or macOS, this process will be relatively straightforward. Microsoft has created a Visual Studio Code Java Pack installer that already includes all relevant components.

1. Download and run the appropriate installer for your operating system.1

Windows: https://aka.ms/vscode-java-installer-win

macOS: https://aka.ms/vscode-java-installer-mac

Linux

If you are using Linux, you should follow the instructions for macOS. However, rather than installing a JDK from a website, please use your distribution's package manager to install it.

Appendix 4: Visualize the schedule

Step 1: Use the data in the "Room.csv" to define the set of provided classrooms as arrays of javascript objects. Example below:

```
var classroomData = [{
    text: "Classroom 1",
    id: " Classroom 1",
    capacity: 30
}, {
    text: " Classroom 9",
    id: " Classroom 9",
    capacity: 30
}, {
    text: " Classroom 23",
    id: " Classroom 23",
    capacity: 30
}, {
    text: " Classroom 45",
    id: " Classroom 1,
    capacity: 30
},
…
}
```

Step 2: Use the data in the "Course.csv" to define the list of courses in the format of arrays of javascript objects

```
var courseData = [{
    id: "ART101",
    text: "Introduction to Visual Studies",
    instructor: "Tram Luong",
    duration: 90,
    color: "#cb6bb2"
  }, {
    id: "CO          RE202",
    text: "Global Humanities and Social Change",
    instructor: "Mathew McDonald",
    duration: 60,
    color: "#56ca85"
  }
]
```

Step 3: We try to convert the output from the algorithm into arrays of javascript objects to be inserted into ***main.js***
Example:

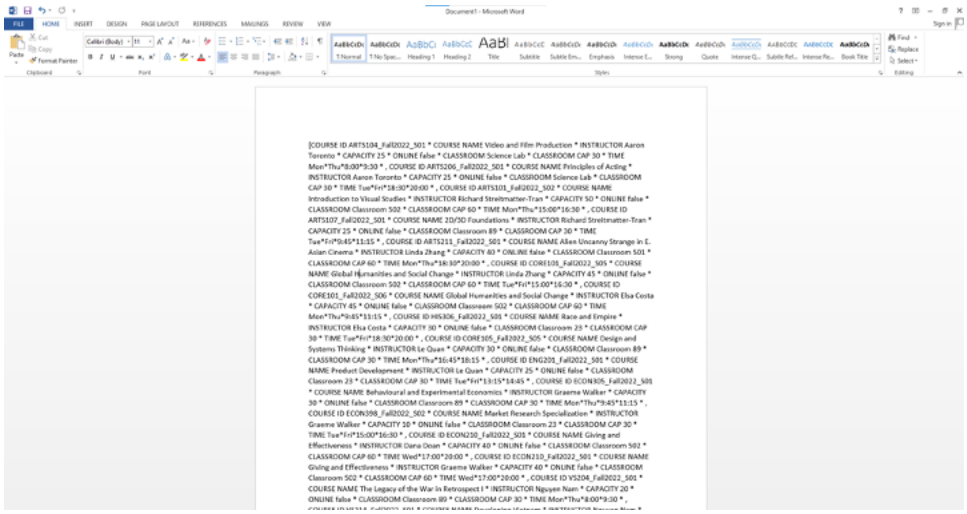
```
var data = [
   {
   classroomId: 'Maker Space',
   courseId: 'ARTS107_Fall2022_S01',
   startDate: new Date(2022, 02, 15, 7, 00),
   endDate: new Date(2022, 02, 15, 10, 45),
   capacity: 30
},

{
   classroomId: 'Maker Space',
   courseId: 'CORE101_Fall2022_S05',
   startDate: new Date(2022, 02, 16, 7, 00),
   endDate: new Date(2022, 02, 16, 10, 45),
   capacity: 30
},
 ]
```

Therefore, for a course taught on 2 days, we considered two distinct objects in data[]
For example: Course CORE101_Fall2022_S05 taught on ***Tuesday*** and ***Thursday*** from 7:00 to 10:45 we will have

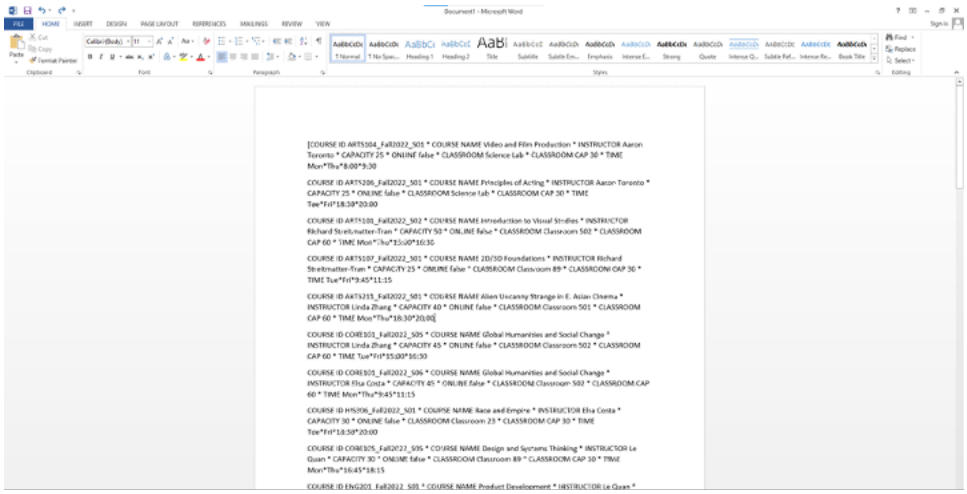
```
var data = [
   {
   classroomId: 'Maker Space',
   courseId: 'CORE101_Fall2022_S05',
   startDate: new Date(2022, 02, 15, 7, 00),
   endDate: new Date(2022, 02, 15, 10, 45),
   capacity: 30
},
{
   classroomId: 'Maker Space',
   courseId: 'CORE101_Fall2022_S05',
   startDate: new Date(2022, 02, 18, 7, 00),
   endDate: new Date(2022, 02, 18, 10, 45),
   capacity: 30
},
 ]
```

Now, the question is how to process the data from running the code in the main section.

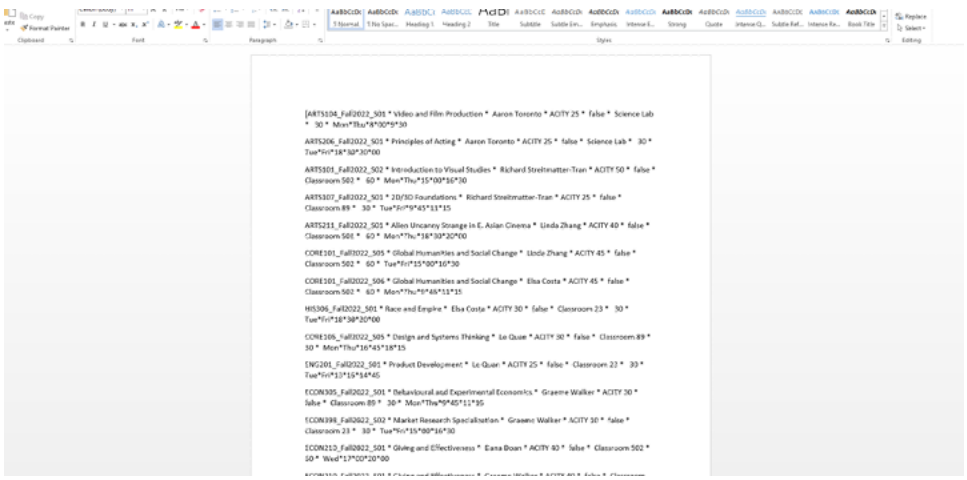
1. Paste the output into Word



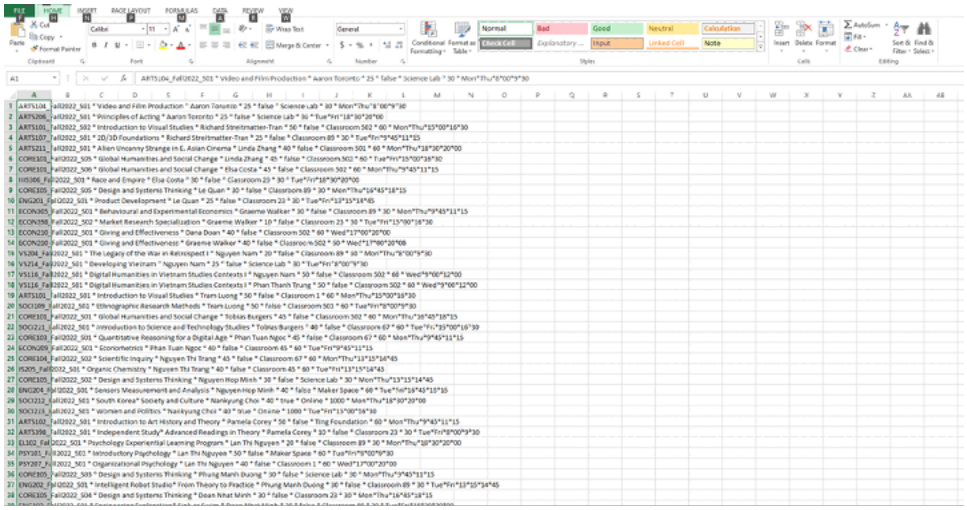
2. Separate each item by line



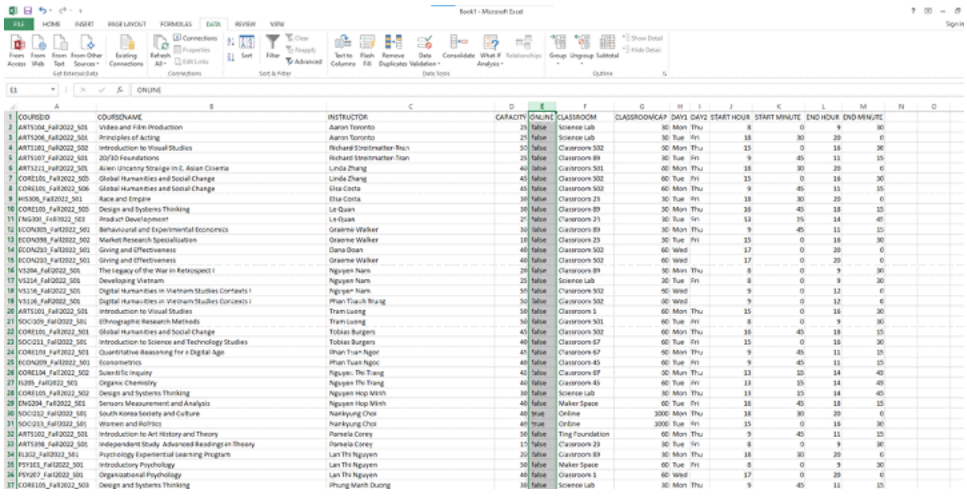
3. Separate the time by ‘ * ’ instead of ‘ : ‘
4. Remove the categories: COURSE ID, COURSE NAME, INSTRUCTOR, CLASROOM, CLASSROOM CAP, TIME



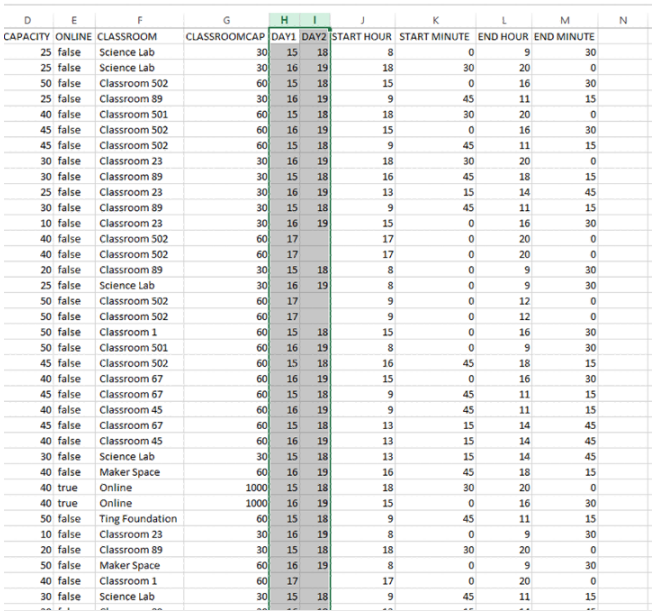
5. Paste into excel



6. Text to column, deliminated with ‘ * ’



7. Denote day 1 and day 2 to number (With Mon = 15, Tue = 16, Wed = 17, Thu = 18, Fri = 19)



D	E	F	G	H	I	J	K	L	M	N
CAPACITY	ONLINE	CLASSROOM	CLASSROOMCAP	DAY1	DAY2	START HOUR	START MINUTE	END HOUR	END MINUTE	
25	false	Science Lab	30	15	18	8	0	9	30	
25	false	Science Lab	30	16	19	18	30	20	0	
50	false	Classroom 502	60	15	18	15	0	16	30	
25	false	Classroom 89	30	16	19	9	45	11	15	
40	false	Classroom 501	60	15	18	18	30	20	0	
45	false	Classroom 502	60	16	19	15	0	16	30	
45	false	Classroom 502	60	15	18	9	45	11	15	
30	false	Classroom 23	30	16	19	18	30	20	0	
30	false	Classroom 89	30	15	18	16	45	18	15	
25	false	Classroom 23	30	16	19	13	15	14	45	
30	false	Classroom 89	30	15	18	9	45	11	15	
10	false	Classroom 23	30	16	19	15	0	16	30	
40	false	Classroom 502	60	17		17	0	20	0	
40	false	Classroom 502	60	17		17	0	20	0	
20	false	Classroom 89	30	15	18	8	0	9	30	
25	false	Science Lab	30	16	19	8	0	9	30	
50	false	Classroom 502	60	17		9	0	12	0	
50	false	Classroom 502	60	17		9	0	12	0	
50	false	Classroom 1	60	15	18	15	0	16	30	
50	false	Classroom 501	60	16	19	8	0	9	30	
45	false	Classroom 502	60	15	18	16	45	18	15	
40	false	Classroom 67	60	16	19	15	0	16	30	
45	false	Classroom 67	60	15	18	9	45	11	15	
40	false	Classroom 45	60	16	19	9	45	11	15	
45	false	Classroom 67	60	15	18	13	15	14	45	
40	false	Classroom 45	60	16	19	13	15	14	45	
30	false	Science Lab	30	15	18	13	15	14	45	
40	false	Maker Space	60	16	19	16	45	18	15	
40	true	Online	1000	15	18	18	30	20	0	
40	true	Online	1000	16	19	15	0	16	30	
50	false	Ting Foundation	60	15	18	9	45	11	15	
10	false	Classroom 23	30	16	19	8	0	9	30	
20	false	Classroom 89	30	15	18	18	30	20	0	
50	false	Maker Space	60	16	19	8	0	9	30	
40	false	Classroom 1	60	17		17	0	20	0	
30	false	Science Lab	30	15	18	9	45	11	15	

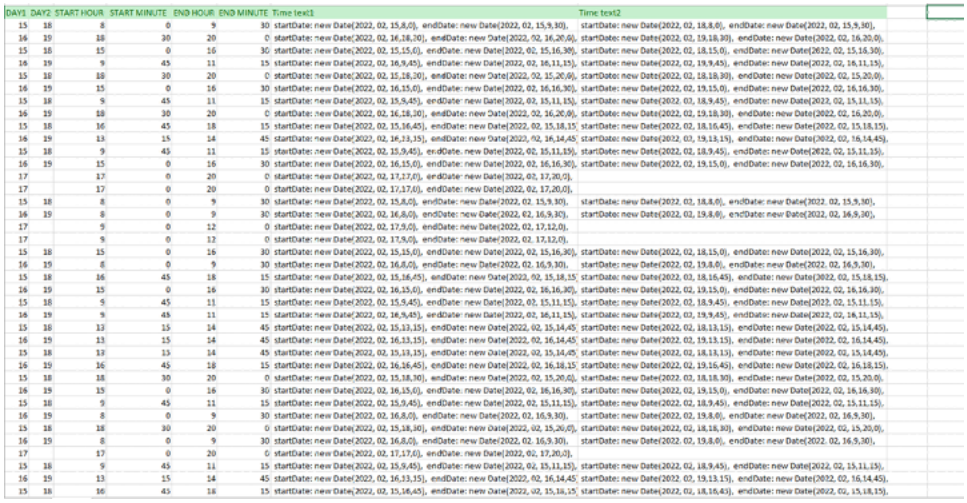
8. Process the time for the 02 days in each course into the format of

“

startDate: new Date(2022, 02, 16, 7, 00),
endDate: new Date(2022, 02, 19, 10, 45),

“

(if the course is taught on 01 days only, then we do not proceed with the time formatting for the second day)



DAY1	DAY2	START HOUR	START MINUTE	END HOUR	END MINUTE	Time text1	Time text2
15	18	8	0	9	30	startDate: new Date(2022, 02, 15,8,0), endDate: new Date(2022, 02, 15,9,30),	startDate: new Date(2022, 02, 18,8,0), endDate: new Date(2022, 02, 15,9,30),
16	19	18	30	20	0	startDate: new Date(2022, 02, 16,18,30), endDate: new Date(2022, 02, 16,20,0),	startDate: new Date(2022, 02, 19,18,30), endDate: new Date(2022, 02, 16,20,0),
15	18	15	0	16	30	startDate: new Date(2022, 02, 15,15,0), endDate: new Date(2022, 02, 15,16,30),	startDate: new Date(2022, 02, 18,15,0), endDate: new Date(2022, 02, 15,16,30),
16	19	9	45	11	15	startDate: new Date(2022, 02, 16,9,45), endDate: new Date(2022, 02, 16,11,15),	startDate: new Date(2022, 02, 19,9,45), endDate: new Date(2022, 02, 16,11,15),
15	18	18	30	20	0	startDate: new Date(2022, 02, 15,18,30), endDate: new Date(2022, 02, 15,20,0),	startDate: new Date(2022, 02, 18,18,30), endDate: new Date(2022, 02, 15,20,0),
16	19	15	0	16	30	startDate: new Date(2022, 02, 16,15,0), endDate: new Date(2022, 02, 16,16,30),	startDate: new Date(2022, 02, 19,15,0), endDate: new Date(2022, 02, 16,16,30),
15	18	9	45	11	15	startDate: new Date(2022, 02, 15,9,45), endDate: new Date(2022, 02, 15,11,15),	startDate: new Date(2022, 02, 18,9,45), endDate: new Date(2022, 02, 15,11,15),
16	19	18	30	20	0	startDate: new Date(2022, 02, 16,18,30), endDate: new Date(2022, 02, 16,20,0),	startDate: new Date(2022, 02, 19,18,30), endDate: new Date(2022, 02, 16,20,0),
15	18	16	45	18	15	startDate: new Date(2022, 02, 15,16,45), endDate: new Date(2022, 02, 15,18,15	startDate: new Date(2022, 02, 18,16,45), endDate: new Date(2022, 02, 15,18,15),
16	19	13	15	14	45	startDate: new Date(2022, 02, 16,13,15), endDate: new Date(2022, 02, 16,14,45	startDate: new Date(2022, 02, 19,13,15), endDate: new Date(2022, 02, 16,14,45),
15	18	9	45	11	15	startDate: new Date(2022, 02, 15,9,45), endDate: new Date(2022, 02, 15,11,15),	startDate: new Date(2022, 02, 18,9,45), endDate: new Date(2022, 02, 15,11,15),
16	19	15	0	16	30	startDate: new Date(2022, 02, 16,15,0), endDate: new Date(2022, 02, 16,16,30),	startDate: new Date(2022, 02, 19,15,0), endDate: new Date(2022, 02, 16,16,30),
17		17	0	20	0	startDate: new Date(2022, 02, 17,17,0), endDate: new Date(2022, 02, 17,20,0),	
17		17	0	20	0	startDate: new Date(2022, 02, 17,17,0), endDate: new Date(2022, 02, 17,20,0),	
15	18	8	0	9	30	startDate: new Date(2022, 02, 15,8,0), endDate: new Date(2022, 02, 15,9,30),	startDate: new Date(2022, 02, 18,8,0), endDate: new Date(2022, 02, 15,9,30),
16	19	8	0	9	30	startDate: new Date(2022, 02, 16,8,0), endDate: new Date(2022, 02, 16,9,30),	startDate: new Date(2022, 02, 19,8,0), endDate: new Date(2022, 02, 16,9,30),
17		9	0	12	0	startDate: new Date(2022, 02, 17,9,0), endDate: new Date(2022, 02, 17,12,0),	
17		9	0	12	0	startDate: new Date(2022, 02, 17,9,0), endDate: new Date(2022, 02, 17,12,0),	
15	18	15	0	16	30	startDate: new Date(2022, 02, 15,15,0), endDate: new Date(2022, 02, 15,16,30),	startDate: new Date(2022, 02, 18,15,0), endDate: new Date(2022, 02, 15,16,30),
16	19	8	0	9	30	startDate: new Date(2022, 02, 16,8,0), endDate: new Date(2022, 02, 16,9,30),	startDate: new Date(2022, 02, 19,8,0), endDate: new Date(2022, 02, 16,9,30),
15	18	16	45	18	15	startDate: new Date(2022, 02, 15,16,45), endDate: new Date(2022, 02, 15,18,15	startDate: new Date(2022, 02, 18,16,45), endDate: new Date(2022, 02, 15,18,15),
16	19	15	0	16	30	startDate: new Date(2022, 02, 16,15,0), endDate: new Date(2022, 02, 16,16,30),	startDate: new Date(2022, 02, 19,15,0), endDate: new Date(2022, 02, 16,16,30),
15	18	5	45	11	15	startDate: new Date(2022, 02, 15,9,45), endDate: new Date(2022, 02, 15,11,15),	startDate: new Date(2022, 02, 18,9,45), endDate: new Date(2022, 02, 15,11,15),
16	19	9	45	11	15	startDate: new Date(2022, 02, 16,9,45), endDate: new Date(2022, 02, 16,11,15),	startDate: new Date(2022, 02, 19,9,45), endDate: new Date(2022, 02, 16,11,15),
15	18	13	15	14	45	startDate: new Date(2022, 02, 15,13,15), endDate: new Date(2022, 02, 15,14,45	startDate: new Date(2022, 02, 18,13,15), endDate: new Date(2022, 02, 15,14,45),
16	19	13	15	14	45	startDate: new Date(2022, 02, 16,13,15), endDate: new Date(2022, 02, 16,14,45	startDate: new Date(2022, 02, 19,13,15), endDate: new Date(2022, 02, 16,14,45),
15	18	13	15	14	45	startDate: new Date(2022, 02, 15,13,15), endDate: new Date(2022, 02, 15,14,45	startDate: new Date(2022, 02, 18,13,15), endDate: new Date(2022, 02, 15,14,45),
16	19	16	45	18	15	startDate: new Date(2022, 02, 16,16,45), endDate: new Date(2022, 02, 16,18,15	startDate: new Date(2022, 02, 19,16,45), endDate: new Date(2022, 02, 16,18,15),
15	18	18	30	20	0	startDate: new Date(2022, 02, 15,18,30), endDate: new Date(2022, 02, 15,20,0),	startDate: new Date(2022, 02, 18,18,30), endDate: new Date(2022, 02, 15,20,0),
16	19	15	0	16	30	startDate: new Date(2022, 02, 16,15,0), endDate: new Date(2022, 02, 16,16,30),	startDate: new Date(2022, 02, 19,15,0), endDate: new Date(2022, 02, 16,16,30),
15	18	9	45	11	15	startDate: new Date(2022, 02, 15,9,45), endDate: new Date(2022, 02, 15,11,15),	startDate: new Date(2022, 02, 18,9,45), endDate: new Date(2022, 02, 15,11,15),
16	19	8	0	9	30	startDate: new Date(2022, 02, 16,8,0), endDate: new Date(2022, 02, 16,9,30),	startDate: new Date(2022, 02, 19,8,0), endDate: new Date(2022, 02, 16,9,30),
15	18	18	30	20	0	startDate: new Date(2022, 02, 15,18,30), endDate: new Date(2022, 02, 15,20,0),	startDate: new Date(2022, 02, 18,18,30), endDate: new Date(2022, 02, 15,20,0),
16	19	8	0	9	30	startDate: new Date(2022, 02, 16,8,0), endDate: new Date(2022, 02, 16,9,30),	startDate: new Date(2022, 02, 19,8,0), endDate: new Date(2022, 02, 16,9,30),
17		17	0	20	0	startDate: new Date(2022, 02, 17,17,0), endDate: new Date(2022, 02, 17,20,0),	
15	18	9	45	11	15	startDate: new Date(2022, 02, 15,9,45), endDate: new Date(2022, 02, 15,11,15),	startDate: new Date(2022, 02, 18,9,45), endDate: new Date(2022, 02, 15,11,15),
16	19	13	15	14	45	startDate: new Date(2022, 02, 16,13,15), endDate: new Date(2022, 02, 16,14,45	startDate: new Date(2022, 02, 19,13,15), endDate: new Date(2022, 02, 16,14,45),
15	18	16	45	18	15	startDate: new Date(2022, 02, 15,16,45), endDate: new Date(2022, 02, 15,18,15	startDate: new Date(2022, 02, 18,16,43), endDate: new Date(2022, 02, 15,18,15),

9. Construct a string representing each item in data[]

“

{

classroomId: 'Maker Space',

courseId: 'CORE101_Fall2022_S05’,

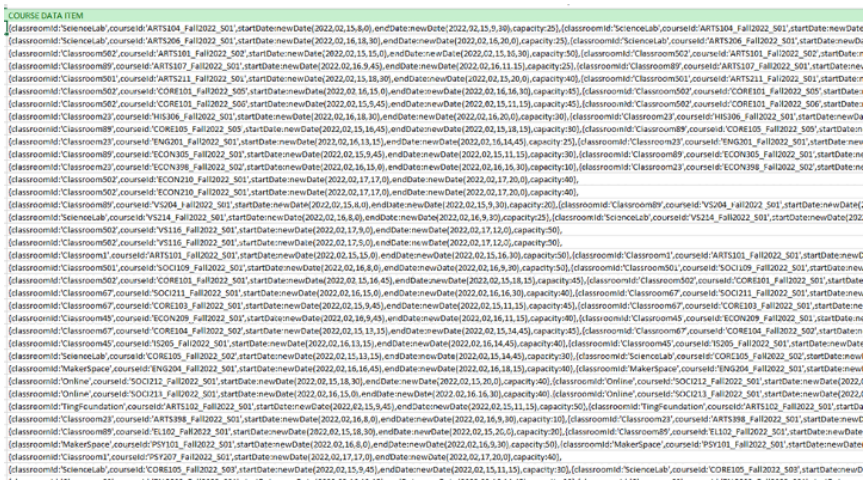
startDate: new Date(2022, 02, 15, 7, 00),

endDate: new Date(2022, 02, 15, 10, 45),

capacity: 30

},

”



10. Concat all the items to create the data[] string of

Var data = [

{classroomId:'MakerSpace',courseId:'CORE104_Fall2022_S03',startDate:newDate(2022,02,15,13,15),endDate:newDate(2022,02,15,14,45),capacity:45},

{classroomId:'MakerSpace',courseId:'CORE104_Fall2022_S03',startDate:newDate(2022,02,18,13,15),endDate:newDate(2022,02,15,14,45),capacity:45},

{classroomId:'Classroom501',courseId:'CORE104_Fall2022_S04',startDate:newDate(2022,02,15,9,45),endDate:newDate(2022,02,15,11,15),capacity:45},

{classroomId:'Classroom501',courseId:'CORE104_Fall2022_S04',startDate:newDate(2022,02,18,9,45),endDate:newDate(2022,02,15,11,15),capacity:45},

...

]

11. Beautify the code:

```
Var data = [
   {
      classroomId: 'MakerSpace',
      courseId: 'CORE104_Fall2022_S03',
      startDate: newDate(2022, 02, 15, 13, 15),
      endDate: newDate(2022, 02, 15, 14, 45),
      capacity: 45
   }, {
      classroomId: 'MakerSpace',
      courseId: 'CORE104_Fall2022_S03',
      startDate: newDate(2022, 02, 18, 13, 15),
      endDate: newDate(2022, 02, 15, 14, 45),
      capacity: 45
   }, {
      classroomId: 'Classroom501',
      courseId: 'CORE104_Fall2022_S04',
      startDate: newDate(2022, 02, 15, 9, 45),
      endDate: newDate(2022, 02, 15, 11, 15),
      capacity: 45
   }, {
      classroomId: 'Classroom501',
      courseId: 'CORE104_Fall2022_S04',
      startDate: newDate(2022, 02, 18, 9, 45),
      endDate: newDate(2022, 02, 15, 11, 15),
      capacity: 45
   },
   ...
]
```

And that is the final data file to be inserted into main.js

Then run the index.html
The full visualized schedule will be shown there.
Sample page: FULBRIGHT SCHEDULE (asthedic.space)