CSC4202

TuNa

ALLOCATING SUITABLE TIME SLOTS FOR A REPLACEMENT CLASS

TEAM MEMBERS

Muhammad Najwan Bin Nizarimi

209897@student.upm.edu.my

Bachelor of Software Engineering

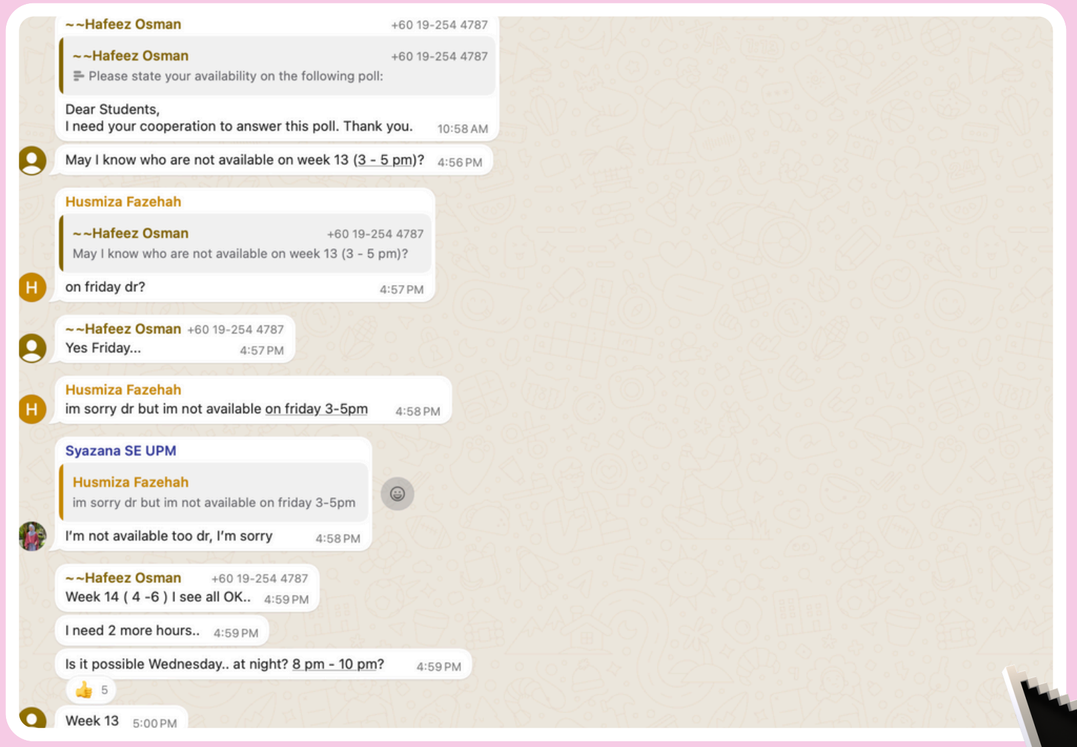
Tuan Nurhusmiza Fazehah Binti Tuan Mat Zawai

209861@student.upm.edu.my

Bachelor of Software Engineering

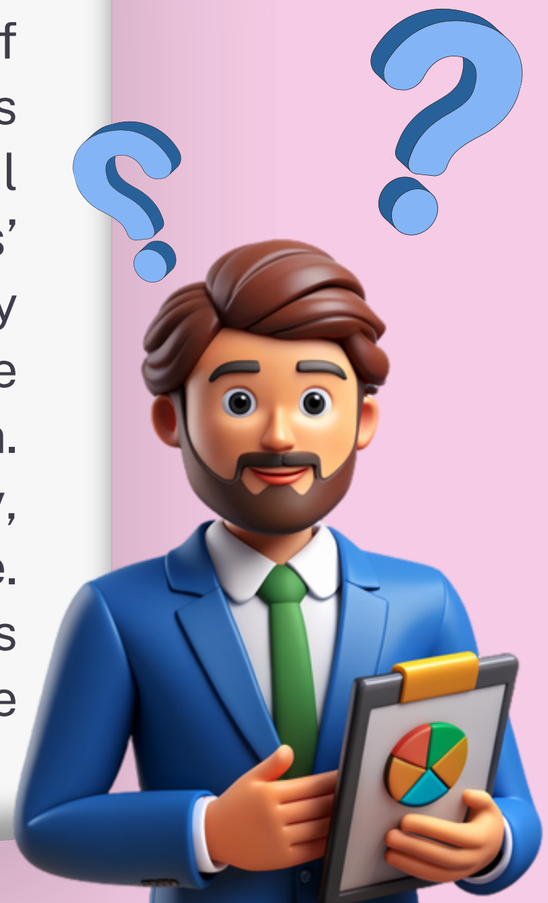
SCENARIO

The illustration below represents students from the Secure Software Development (SWE4356) class at University Putra Malaysia (UPM), where the lecturer struggles to find replacement classes that accommodate all the students



PROBLEM DEFINITION

Dr. Hafeez is a dedicated lecturer in the Department of Software Engineering at Universiti Putra Malaysia (UPM), located in Serdang, Malaysia. Throughout the semester, his busy schedule and multiple public holidays occurring on class days have resulted in multiple postponed lectures. This has led to inconsistent student attendance and the risk of important topics being left uncovered before the final exams. To address these challenges, Dr. Hafeez seeks a solution to identify the most optimal time slots for replacement classes by analysing both his and his students' schedules. The proposed algorithm promises efficiency by quickly pinpointing the best time slots, accuracy by ensuring maximum possible attendance, and convenience by reducing the manual effort in coordination. The goal is to ensure that all missed lectures are rescheduled effectively, maximising student attendance and ensuring complete syllabus coverage. The expected output is a list of optimal time slots for replacement classes without conflicts. This solution aims to make sure all important lectures are completed and students are well-prepared for their final exams.



IMPORTANCES

- **Maintaining Academic Standards**

Each subject has designated credit hours and content that must be completed to meet educational standards.

- **Avoiding Scheduling Conflicts**

Replacement classes might be scheduled at inconvenient times which lead to conflicts with other classes or personal commitments.

- **Ensuring Fairness**

Ensure all students receive the same amount of learning time, as it can be unfair if certain students miss out on certain topics due to time constraints.

- **Minimising Disruption**

Properly scheduled replacement classes can help minimise additional disruptions to the academic calendar and ensure a smooth learning experience.

DEVELOPMENT OF THE MODEL

- **Data used**: Empty or available time slots from the lecture's and students' schedules for SWE3456 class
- **Objective**: To cover missed lecture hours during the holiday

CONSTRAINTS AND REQUIREMENTS

- Only on weekdays from 8 a.m to 6 p.m.
- Can be held online or physically
- Must cover for 2 hours
- Must not overlap with other scheduled classes
- A person is considered free when they do not have any classes during that time

DATA TYPE

Lecturer

TIME/DAY	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18		
MONDAY							SWE4356 (GROUP 1)	SWE4356 (GROUP 1)				
TUESDAY				SWE4356 (GROUP 1)								
WEDNESDAY							SWE4356 (GROUP 2)	SWE4356 (GROUP 2)				
THURSDAY			SWE4356 (GROUP 2)									
FRIDAY												
SATURDAY												
SUNDAY												

Students

Date : 17/24/2024

ACADEMIC DIVISION
UNIVERSITI PUTRA MALAYSIA

TIME TABLE
SECOND SEMESTER 2023/2024

NAME : HAWITH DZIKRI BIN HASHIM
MATRIC NO. : 210183
FACULTY : FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
PROGRAMME : 46D-2 BACHELOR OF SOFTWARE ENGINEERING WITH HONOURS
MAJOR : NO MAJOR
MINOR : NO MINOR

TIME / DAY	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23
MONDAY			SSE3400 (GROUP 3) BD-05 (LECTURE)		SWE4357 (GROUP 1) BKDFTAKTM (LECTURE)		SWE4356 (GROUP 1) A2-0607 (LECTURE)	SWE4356 (GROUP 1) A2-0607 (LECTURE)	CSC4456 (GROUP 1) MAK.PEND ANA FSKTM (LECTURE)	CSC4456 (GROUP 1) MAK.PEND ANA FSKTM (LECTURE)					
TUESDAY	SSE3400 (GROUP 2) MAK.KP2 FSKTM (LAB)	SSE3400 (GROUP 2) MAK.KP2 FSKTM (LAB)	SSE3400 (GROUP 2) MAK.KP2 FSKTM (LAB)	SWE4356 (GROUP 1) A1-01 (LECTURE)			SWE4357 (GROUP 1) A2-0607 (LECTURE)	SWE4357 (GROUP 1) A1-01 (LECTURE)	CND3102 (GROUP 2) BKDFTAKTM (LECTURE)	CND3102 (GROUP 2) BKDFTAKTM (LECTURE)					
WEDNESDAY	CSC4456 (GROUP 1) MAK.PEND ANA FSKTM (LECTURE)	CSC4456 (GROUP 1) MAK.PEND ANA FSKTM (LECTURE)	CEL2105 (GROUP 1) DKB (LECTURE)	CEL2105 (GROUP 1) DKB (LECTURE)											
THURSDAY	CND3102 (GROUP 1) MAK.NANG KAJIAN FSKTM2 (LAB)	CND3102 (GROUP 1) MAK.NANG KAJIAN FSKTM2 (LAB)	CND3102 (GROUP 1) MAK.NANG KAJIAN FSKTM2 (LAB)												
FRIDAY			SSE3400 (GROUP 2) MAK.KP2 FSKTM (LAB)	SSE3400 (GROUP 2) MAK.KP2 FSKTM (LAB)											
SATURDAY															
SUNDAY															

Printed by : HAWITH DZIKRI BIN HASHIM
Code : 16503

Date : 26/24/2024

ACADEMIC DIVISION
UNIVERSITI PUTRA MALAYSIA

TIME TABLE
SECOND SEMESTER 2023/2024

NAME : XUE RUI
MATRIC NO. : 209761
FACULTY : FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
PROGRAMME : 46D-2 BACHELOR OF SOFTWARE ENGINEERING WITH HONOURS
MAJOR : NO MAJOR
MINOR : NO MINOR

TIME / DAY	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23
MONDAY			SSE3400 (GROUP 1) BD-04 (LECTURE)		SWE4357 (GROUP 2) A2-09 (LECTURE)		SWE4356 (GROUP 1) A2-0607 (LECTURE)	SWE4356 (GROUP 1) A2-0607 (LECTURE)							
TUESDAY	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SWE4356 (GROUP 1) A1-01 (LECTURE)			SWE4357 (GROUP 1) A2-0607 (LECTURE)	SWE4357 (GROUP 1) A1-01 (LECTURE)							
WEDNESDAY			CEL2105 (GROUP 1) BS-08/09 (LECTURE)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	CEL2105 (GROUP 1) BS-08/09 (LECTURE)		SKM3092 (GROUP 1) M.P.DATA FSKTM (LECTURE)								
THURSDAY		SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)			SKM3092 (GROUP 1) M.P.DATA FSKTM (LECTURE)	SKM3092 (GROUP 1) BKDFTAKTM (LECTURE)							
FRIDAY															
SATURDAY															
SUNDAY															

Printed by : XUE RUI
Code : 16503

Date : 17/24/2024

ACADEMIC DIVISION
UNIVERSITI PUTRA MALAYSIA

TIME TABLE
SECOND SEMESTER 2023/2024

NAME : TUAN NURHUSMIZA FAZEHAH BINTI TUAN MAT ZAWAI
MATRIC NO. : 209961
FACULTY : FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
PROGRAMME : 46D-2 BACHELOR OF SOFTWARE ENGINEERING WITH HONOURS
MAJOR : NO MAJOR
MINOR : NO MINOR

TIME / DAY	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23
MONDAY	SWE4355 (GROUP 1) BKDFTAKTM (LECTURE)	SWE4355 (GROUP 1) BKDFTAKTM (LECTURE)	SSE3400 (GROUP 1) BD-04 (LECTURE)		SWE4357 (GROUP 1) BKDFTAKTM (LECTURE)		SWE4356 (GROUP 1) A2-0607 (LECTURE)	SWE4356 (GROUP 1) A2-0607 (LECTURE)							
TUESDAY	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SWE4356 (GROUP 1) A1-01 (LECTURE)			SWE4357 (GROUP 1) A2-0607 (LECTURE)	SWE4357 (GROUP 1) A1-01 (LECTURE)							
WEDNESDAY			CEL2105 (GROUP 1) DKB (LECTURE)	CEL2105 (GROUP 1) DKB (LECTURE)	SWE4355 (GROUP 1) BKDFTAKTM (LECTURE)		CSC4202 (GROUP 1) BK.OPSKTM (LECTURE)	CSC4202 (GROUP 1) BK.OPSKTM (LECTURE)							
THURSDAY	CSC4202 (GROUP 1) A1-07 (LECTURE)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)	SSE3400 (GROUP 1) MAK.KP1 FSKTM (LAB)											
FRIDAY															
SATURDAY															
SUNDAY															

Printed by : TUAN NURHUSMIZA FAZEHAH BINTI TUAN MAT ZAWAI
Code : 16503

Tarikh : 17/24/2024

BAHAGIAN AKADEMIK
UNIVERSITI PUTRA MALAYSIA

JADUAL WAKTU
SEMESTER KEDUA SESI 2023/2024

NAMA : NUR HANAN BINTI AHMAD SALLEHUDDIN
NO. MATRIK : 211555
FAKULTI : FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT
PROGRAM : 46D-2 BACELOR KEJURUTERAAN PERISIAN DENGAN KEPUJIAN
MAJOR : TIADA MAJOR
MINOR : TIADA MINOR

MASA / HARI	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23
ISNIN	SWE4355 (KUMP 1) BKDFTAKTM) (KULIAH)	SWE4355 (KUMP 1) BKDFTAKTM) (KULIAH)	SSE3400 (KUMP 2) BD-05 (KULIAH)		SWE4357 (KUMP 1) BKDFTAKTM) (KULIAH)		SWE4356 (KUMP 1) A2-0607 (KULIAH)	SWE4356 (KUMP 1) A2-0607 (KULIAH)							
SELASA	SSE3400 (KUMP 1) MAK.KP2 FSKTM (AMALI)	SSE3400 (KUMP 1) MAK.KP2 FSKTM (AMALI)	SSE3400 (KUMP 1) MAK.KP2 FSKTM (AMALI)	SWE4356 (KUMP 1) A1-01 (KULIAH)			SWE4357 (KUMP 1) A2-0607 (KULIAH)	SWE4357 (KUMP 1) A1-01 (KULIAH)	CND3102 (KUMP 2) BKDFTAKTM) (KULIAH)	CND3102 (KUMP 2) BKDFTAKTM) (KULIAH)					
RABU				SWE3307 (KUMP 1) A2-08 (KULIAH)	SWE4355 (KUMP 1) BKDFTAKTM) (KULIAH)										
KHAMIS	CND3102 (KUMP 1) MAK.NANG KAJIAN FSKTM2 (AMALI)	CND3102 (KUMP 1) MAK.NANG KAJIAN FSKTM2 (AMALI)	CND3102 (KUMP 1) MAK.NANG KAJIAN FSKTM2 (AMALI)	SWE3307 (KUMP 1) BS-08/09 (KULIAH)	SWE3307 (KUMP 1) BS-08/09 (KULIAH)										
JUMAAT			SSE3400 (KUMP 2) MAK.KP2 FSKTM (AMALI)	SSE3400 (KUMP 2) MAK.KP2 FSKTM (AMALI)	SSE3400 (KUMP 2) MAK.KP2 FSKTM (AMALI)										
SABTU															
AHAD															

Disahkan Oleh : NUR HANAN BINTI AHMAD SALLEHUDDIN
Code : 16503

ALGORITHMS REVIEW

Sorting

S: Straigthtforward way to organise schedules by date or availability

W: Doesn’t address the complexity of schedules conflicts optimization for maximum attendances

DP

S: Can be used to find optimal time slots for replacement classes by recursively considering sub-problems

W: DP can require significant memory especially with large inputs.

Greedy

S: Easy to implement as it often requires less code due to its straightforward character

W: Do not always produce optimal results especially when future decisions are dependent on earlier decisions

DAC

S: Efficient as breaking down a problem into sub-problems, solve it, and combine it back together can speed up the computation

W: Complicated to implement for problems that highly dependant to other problems

Graph

S: Can represent the relationship between time slots, conflicts and dependencies to identify the time slots for replacement classes

W: Can be used under the assumption of static data and reflects that does not reflect real-word dynamic changes

ALGORITHM SPECIFICATION

Aspect	Greedy	Dynamic Programming
Approach	Makes locally optimal choices at each step without considering future consequences.	Solves problems by breaking them down into smaller overlapping subproblems.
Decision Making	Makes decisions based on immediate gains or local optimal solutions	Evaluates all possible decisions systematically to find the optimal solution
Backtracking	Does not involve backtracking	Does involve backtracking
Complexity	Lower time and space complexity	Higher time and space complexity
Application	Suitable for quick scheduling with fewer constraint	Suitable if multiple constraints and factors need thorough analysis
Examples	Minimum Spanning Tree, Shortest Path algorithms.	Fibonacci sequence, Longest Common Subsequence.

Therefore, Dynamic Programming algorithm is better suited as it provides multiple choices for the students and lecturers, resulting in an optimal allocation of time for replacement classes.

PSEUDOCODE

```
START

DEFINE WORKING_HOURS_START AS 8
DEFINE WORKING_HOURS_END AS 18
DEFINE REQUIRED_FREE_HOURS AS 2

CLASS DPAlgorithm

FUNCTION findContinuousFreeSlot(lecturerAvailability,
studentAvailability)
INITIALIZE dp AS ARRAY OF BOOLEAN WITH LENGTH
(WORKING_HOURS_END - WORKING_HOURS_START + 1)

FOR hour FROM WORKING_HOURS_START TO
WORKING_HOURS_END
IF isSlotAvailable(hour, hour + REQUIRED_FREE_HOURS,
lecturerAvailability)
AND isSlotAvailable(hour, hour +
REQUIRED_FREE_HOURS, studentAvailability)
SET dp[hour - WORKING_HOURS_START] TO TRUE
ELSE
SET dp[hour - WORKING_HOURS_START] TO FALSE
END IF
END FOR

printContinuousFreeSlots(dp)
END FUNCTION

FUNCTION isSlotAvailable(startHour, endHour, availability)
FOR hour FROM startHour TO endHour - 1
IF availability[hour] IS FALSE
RETURN FALSE
END IF
END FOR
RETURN TRUE
END FUNCTION

FUNCTION printContinuousFreeSlots(dp)
PRINT "Free 2-hour slots:"
FOR hour FROM 0 TO LENGTH(dp) - REQUIRED_FREE_HOURS
IF dp[hour] IS TRUE AND dp[hour + 1] IS TRUE
PRINT "Free slot from", WORKING_HOURS_START + hour, "to",
WORKING_HOURS_START + hour + REQUIRED_FREE_HOURS
END IF
END FOR
END FUNCTION

END CLASS

FUNCTION main()
INITIALIZE lecturerAvailability AS ARRAY OF BOOLEAN WITH
LENGTH 24
INITIALIZE studentAvailability AS ARRAY OF BOOLEAN WITH
LENGTH 24

FOR hour FROM 0 TO 23
SET lecturerAvailability[hour] TO TRUE
SET studentAvailability[hour] TO TRUE
END FOR

SET lecturerAvailability[10] TO FALSE
SET studentAvailability[15] TO FALSE

CREATE instance OF DPAlgorithm

CALL instance.findContinuousFreeSlot(lecturerAvailability,
studentAvailability)

END FUNCTION

END
```

ALGORITHM ANALYSIS

ALGORITHM CORRECTNESS:

Analyzed through asymptotic complexity and recurrence relations.

Asymptotic complexity examines runtime behavior as input size grows.

ALGORITHM ANALYSIS

TIME COMPLEXITY

- Overall time complexity: O(d x n).
  - d: number of days.
  - n: number of people.
- Determined by iterating through each day and checking availability for each person within specified hours.
- Linear scan of availability list for each student.
- Combined complexity results in O(d x n).

ALGORITHM ANALYSIS

RECURRENCE RELATIONS

- Specifies how solutions to larger subproblems depend on smaller subproblems.
- Defines solution to a larger problem based on its previous values.
- Relation: ‘dp[i] = lecturerFree && studentsFree’.
- dp[i]: whether a 2-hour free slot starting at hour ‘i’ is feasible.
- Builds on smaller subproblems to solve the entire working period.

ALGORITHM ANALYSIS

IMPLEMENTATION

- Base case: Initialize ‘boolean[] dp = new boolean[WORKING_HOURS_END]’.
  - Serves as the dynamic programming table.
- Inductive step: Solve smaller subproblems to larger ones.
  - In ‘findContinuousFreeSlot’ method:
    - Loop iterates over each hour.
    - Checks if both lecturer and students are available for a 2-hour slot starting at hour ‘i’.
    - Sets ‘dp[i]’ to true if conditions are met, otherwise false.
  - After populating ‘dp’ array:
    - Another loop collects all hours where ‘dp[i]’ is true into ‘freeSlots’ list.
    - Represents valid 2-hour continuous time slots.

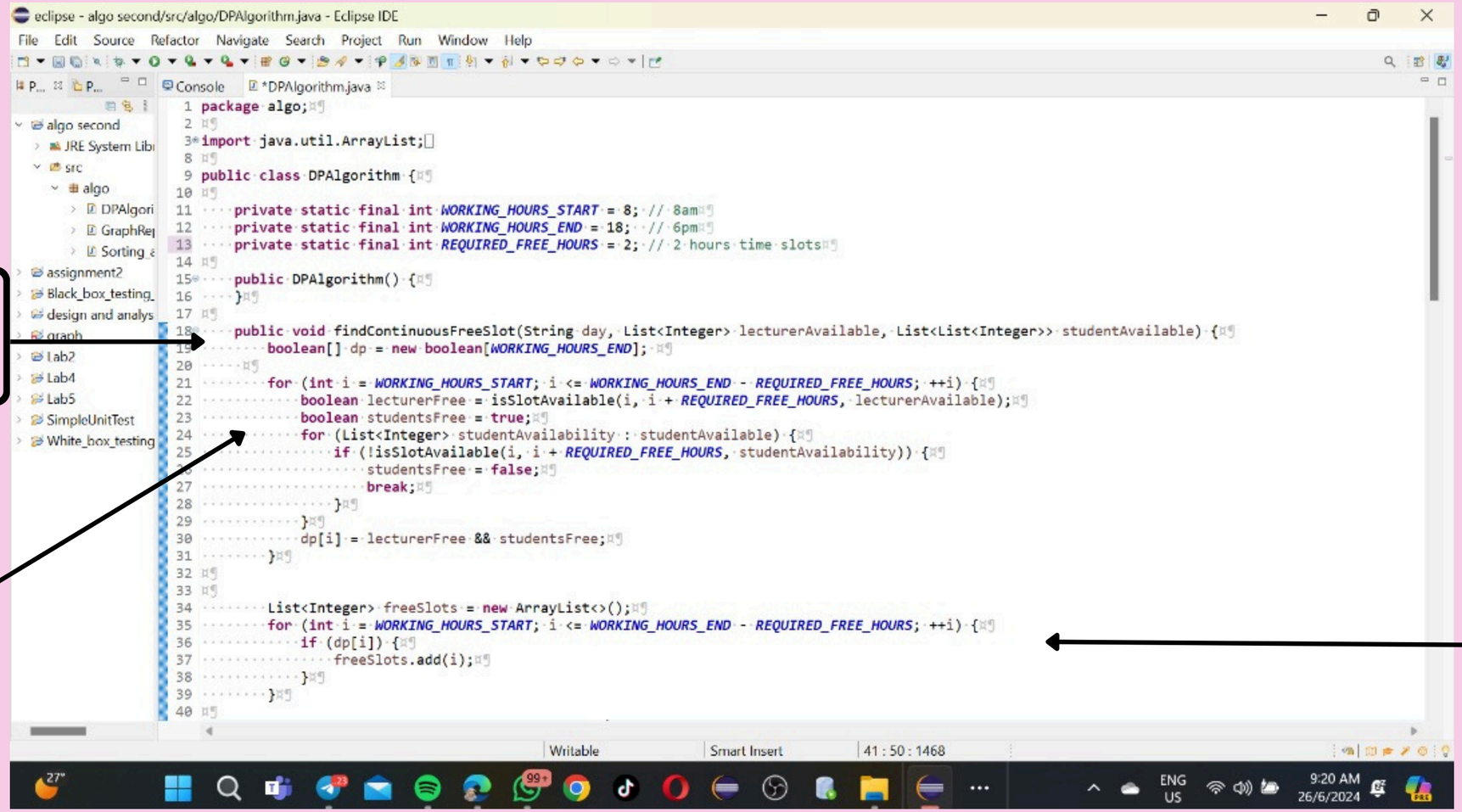
IMPLEMENTATION

Initialisation of ‘dp’ array

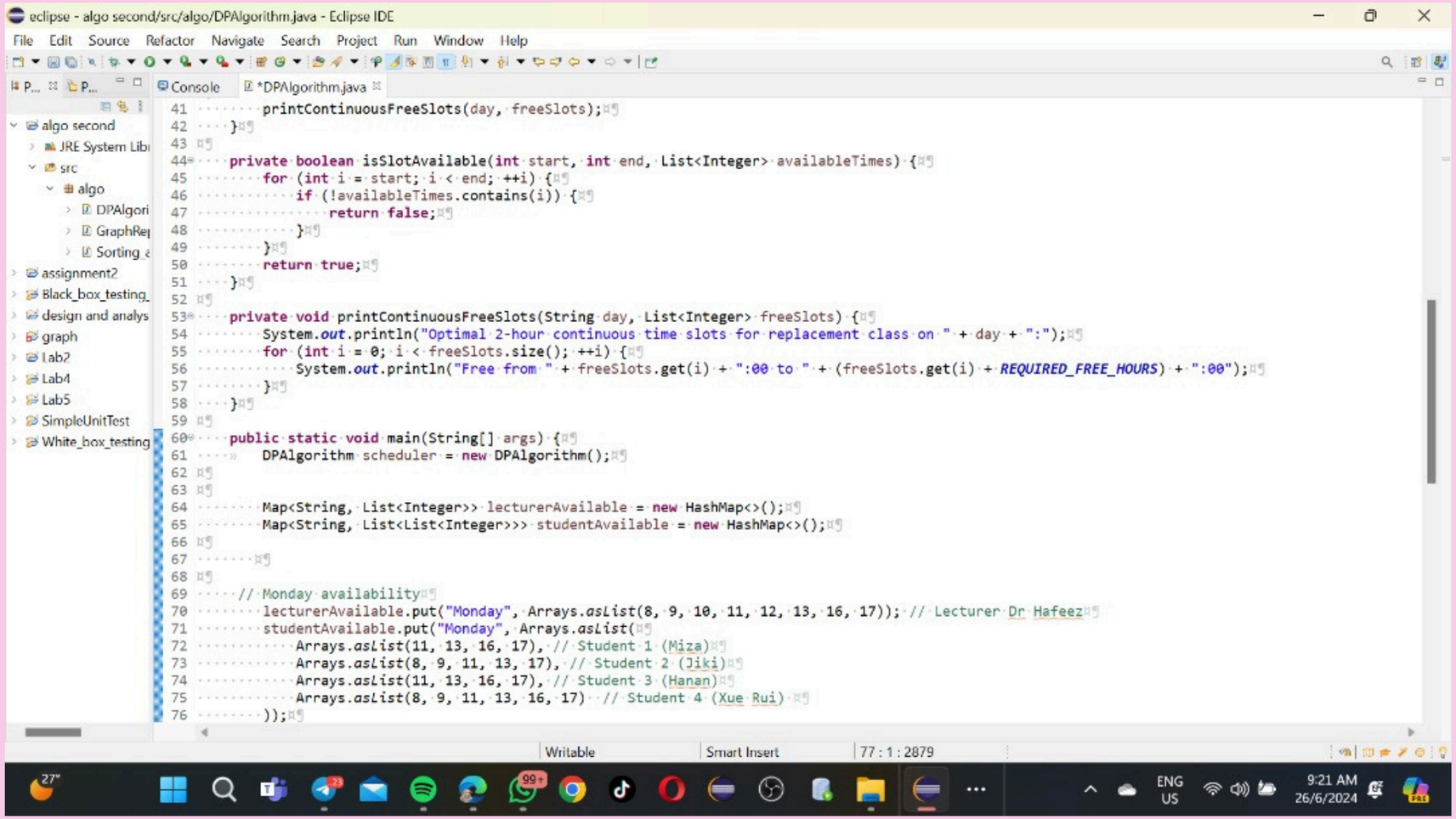
Filling the ‘dp’ array

DP Table Update

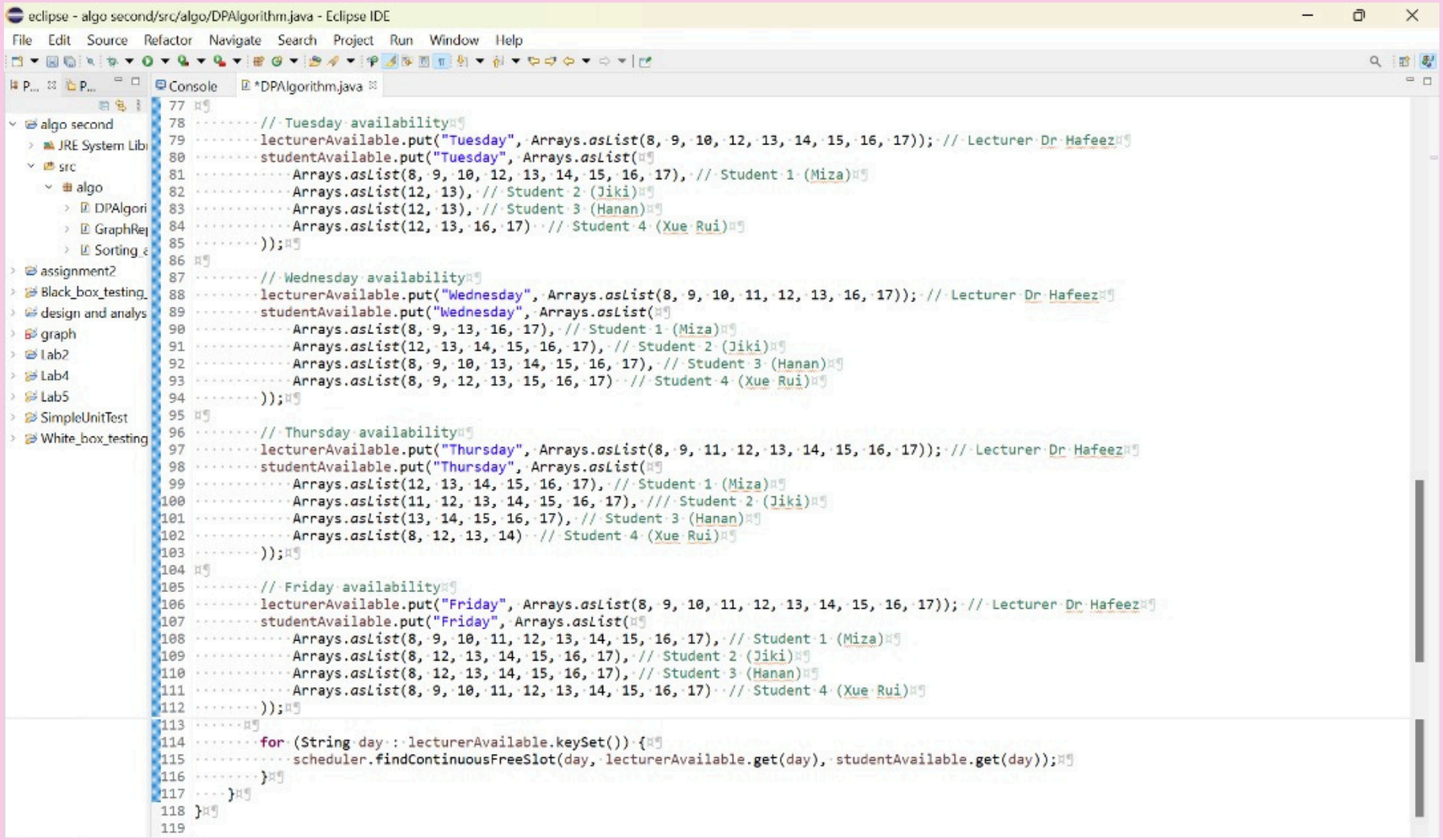
Utilisation of results



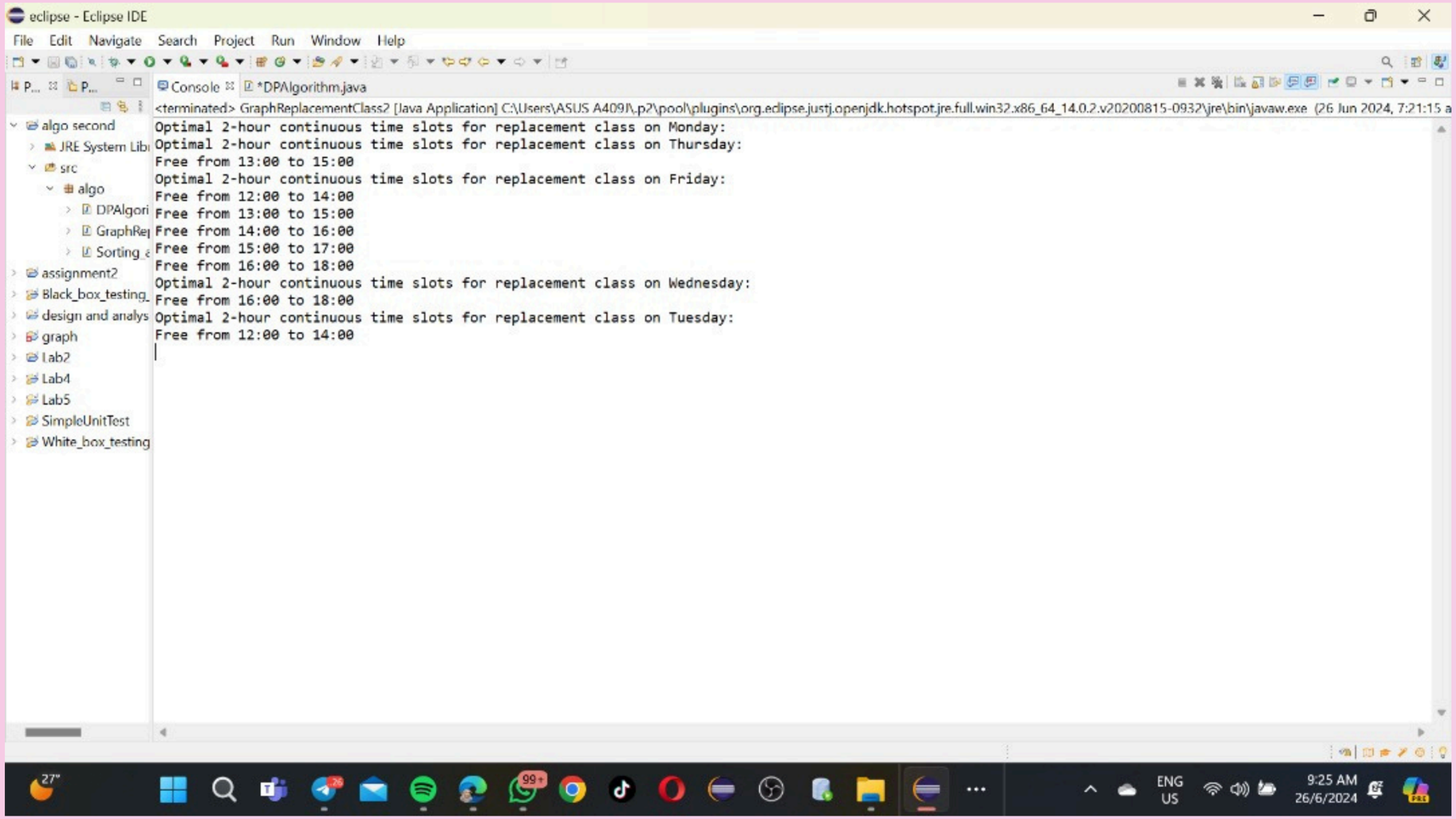
IMPLEMENTATION



IMPLEMENTATION



OUTPUT


eclipse - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Console
*DPAlgorithm.java
<terminated> GraphReplacementClass2 [Java Application] C:\Users\ASUS A409J\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (26 Jun 2024, 7:21:15 a
Optimal 2-hour continuous time slots for replacement class on Monday:
Optimal 2-hour continuous time slots for replacement class on Thursday:
Free from 13:00 to 15:00
Optimal 2-hour continuous time slots for replacement class on Friday:
Free from 12:00 to 14:00
Free from 13:00 to 15:00
Free from 14:00 to 16:00
Free from 15:00 to 17:00
Free from 16:00 to 18:00
Optimal 2-hour continuous time slots for replacement class on Wednesday:
Free from 16:00 to 18:00
Optimal 2-hour continuous time slots for replacement class on Tuesday:
Free from 12:00 to 14:00
algo second
JRE System Libi
src
algo
DPAlgori
GraphRep
Sorting
assignment2
Black_box_testing_
design and analys
graph
Lab2
Lab4
Lab5
SimpleUnitTest
White_box_testing
ENG
US
9:25 AM
26/6/2024

THANK YOU